NASA/TM-1999-208641/VER3/VOL1

ICESat (GLAS) Science Processing Software Document Series

Volume 1
Science Software Management Plan
Version 3.0

David W. Hancock III
Observational Science Branch
Laboratory for Hydrospheric Processes
NASA/GSFC Wallops Flight Facility
Wallops Island, Virginia 23337

August 1998

ICESat Contacts:

H. Jay Zwally, ICESat Project Scientist
NASA Goddard Space Flight Center
Greenbelt, Maryland 20771

Bob E. Schutz, GLAS Science Team Leader
University of Texas Center for Space Research
Austin, Texas 78759-5321

David W. Hancock III, Science Software Development Leader
NASA/GSFC Wallops Flight Facility
Wallops Island, Virginia 23337

Foreword

This document provides the Software Management Plan for the GLAS Standard Data Software (SDS) supporting the GLAS instrument on the EOS ICESat Spacecraft. The SDS encompasses the ICESat Science Investigator-led Processing System (I-SIPS) Software and the Instrument Support Terminal (IST) Software. For the I-SIPS Software, the SDS will produce Level 0, Level 1, and Level 2 data products as well as the associated product quality assessments and descriptive information. For the IST Software, the SDS will accommodate the GLAS instrument support areas of engineering status, command, performance assessment, and instrument health status.

This Management Plan is developed under the structure of the NASA STD-2100-91, a NASA standard defining a four-volume set of documents to cover an entire software life cycle. Under this standard a section of any volume may, if necessary, be rolled out to its own separate document. Within this standard software development structure, this Science Software Management Plan provides the information required by the 1995 EOS Project GLAS statement of work for the deliverable preliminary document "Software Management Plan".

This document was prepared by the Observational Science Branch at NASA Goddard Space Flight Center, Wallops Flight Facility, Wallops Island, Virginia, in support of B. E. Schutz, GLAS Science Team Leader for the GLAS Investigation. This work was performed under the direction of David W. Hancock, III, who may be contacted at (757) 824-1238, hancock@osb.wff.nasa.gov (e-mail), or (757) 824-1036 (fax).

Document Change History

Document Name: GLAS Science Software Management Plan		
Version Number	**Date**	**Nature of Change**
Preliminary	December 31, 1995	Original Version
Updated Preliminary	December 31, 1996	
Version 2.2	July 1998	Update of Terminology and Processes
Version 3.0	August 1998	Minor fixes to Version 2.2; change bars removed.

Table of Contents

Section 6 Development Activities Plan

Section 7 Sustaining Engineering and Operations Activities Plan

Section 8 Assurance Plan

Section 9 Risk Management Plan

Section 10 Configuration Management Plan

Section 11 Delivery and Operational Transition Plan

List of Figures

List of Tables

Preface

The EOS ICESat (Ice, Cloud, and land Elevation Satellite) Spacecraft is part of NASA's Earth Science Enterprise. The ICESat platforms will carry the Geoscience Laser Altimeter System (GLAS) into a nominal 600 kilometer, 94° inclination (non-Sun synchronous), circular orbit. The ICESat series consists of three consecutive launch platforms covering a 15 year mission.

The GLAS laser is a frequency-doubled, cavity-pumped, solid state Nd:YAG laser having a dual energy level, spectral pulse capability of 120 and 60 millijoules for an output wavelength of 1.064 and 0.532 μm, respectively. The infrared and green pulse spectra are used to perform both surface topography and atmospheric measurements. The laser output is generated at a rate of 40 pulses per second with a beam divergence of 0.1 milliradians. Reflected energies from the 70-meter nadir spots are received by an 80-centimeter diameter telescope.

In addition to the laser system, GLAS is composed of star cameras and an external laser pointing monitor for laser pointing reference establishment. The ICESat Spacecraft will carry its own star camera and GPS receivers to aid in location determination.

The progression of data flow for GLAS starts with the raw instrument data recorded and archived by the EOS Data and Operations System (EDOS). The raw data are processed into the GLAS Level 0 Data within EDOS. The GLAS Level 1A, Level 1B, and Level 2 Standard Data Products are produced from the Level 0 Data. The Level 1A and Level 1B Data Products include the 1.064 and 0.532 μm laser return travel times, transmitted and received pulse energies and counts, surface waveform parameters, cloud heights, atmospheric backscatter, height vectors, calibration, external laser pointing monitor data, the star camera data, and the Global Positioning [Satellite] System (GPS) receiver data. The precision orbit/attitude data, and meteorological data (atmospheric profile) derived from the external laser pointing monitor, the star cameras, the GPS receiver, the spacecraft attitude data, and other instrument and tracking data are applied to the geo-located Level 1B Data Products and the Level 2 Data Products.

Within this document, the term "instrument data product" indicates the Level 0 data product generated by EDOS from the raw data product collected onboard the spacecraft and telemetered to the ground receiving station. The term "standard data products" refers to those EOS instrument data products listed in the Earth Science Data and Information System (ESDIS) Project data base that are routinely generated within the ESDIS Distributed Active Archive Center (DAAC) or Science Computing Facilities (SCFs). The GLAS Data Product levels are described more fully in the Glossary of this document. The term "special data products" refers to those EOS instrument-derived products listed in the ESDIS Project database that are generated within Science Computing Facilities on a request basis. Each data product has a unique Product Identification code assigned by the EOS Senior Project Scientist. These data products will have been physically generated as a collection of EOS data parameters in a product aggregate or file. Data parameters will be retrievable from the DAAC. Data parameters are composed of GLAS elements, i.e., data items and arrays of items. The arrays and data items consist of measured or derived instrument values.

Items to be Resolved

1) Section 6.3 describes the formal reviews f or the SDS software. The Blue Book asks for the following to be described in this document:

    - How reviews are conducted
    - Who attends
    - discrepancy handling
    - decision process

    Is this information appropriate for this document? Is there a standard or method for conducting reviews or do we have to come up with something for this document?

2) For Configuration Management, need to review ClearCase configuration status products/reports and compare to what is proposed.

3) Need to update development schedule.

Section 1

Introduction

1.1 Identification of Document

This is the Management Plan for the development of the GLAS Standard Data Software (SDS). The unique identification number within the GLAS Standard Data Software document numbering scheme is GLAS-SMP-1100. Successive editions of this document will be uniquely identified by version number and by the cover and page date marks.

1.2 Scope of Document

This software management plan will be used by the GLAS Science Team Leader to manage the development of the GLAS Standard Data Software. This document will be used by the GLAS SDS Development Team as guidelines for GLAS SDS development. This document is generated by members of the GLAS SDS Development Team in support of the GLAS Science Team Leader. The GLAS Science Team Leader is responsible for the development of the SDS. Each GLAS Science Team member is responsible to the Science Team Leader to support this requirement. The GLAS SDS Development Team is responsible to the Science Team Leader to define, develop and deliver the software defined in this Management Plan.

1.3 Purpose and Objectives of Document

The purpose of this Management Plan is to define the methods and schedules to develop the GLAS SDS. This plan defines all the major items to be delivered and provides their delivery schedule. The methods and approaches to be used during the development process are described. This plan defines specific end items that can be used to track and verify progress of the development effort.

This plan is designed to be used throughout the life of the development process to allow early identification of problems. It defines methods to identify and to resolve problems so that excellent software that performs all required functions, together with its documentation, will be delivered on schedule. This Management Plan is to provide the GLAS Science Team Leader, the GLAS Instrument Team Leader, the ESDIS Project, and the GLAS SDS Development Team with the information needed to successfully manage the development of all required software and documentation for the GLAS SDS, and to deliver the products in a timely manner.

1.4 Document Status and Schedule

This is version 2.2 of the GLAS Science Software Management Plan. Revisions to this document will be made and released as necessary.

1.5 Document Organization

The GLAS Science Software Management Plan structure is based on the document organization for the Management Plan Volume under the NASA Software Documentation Standard - Software Engineering Program [Reference 2.2c]. Table 1-1 "The Science Software Management Plan Within the Standard Documentation Series" depicts the Science Software Management Plan as a part of the first Volume among the four volumes defined under the NASA Software Documentation Standards. Figure 1-1 "GLAS SDS Documentation Tree" on page 1-3 shows the relationship of the Science Software Management Plan to other deliverable science investigation documents.

Table 1-1 The Science Software Management Plan Within the Standard Documentation Series

GLAS Science Software	
Management Plan Volume	GLAS Science Software Management Plan
	GLAS Science Data Management Plan
Product Specification Volume	GLAS Science Software Requirements
	GLAS Science Software Architectural Design
	GLAS Science Software Detailed Design
	GLAS Science Software User's Guide/Operational Procedures Manual
	GLAS Science Software Version Description
Assurance and Test Procedures Volume	GLAS Science Software Assurance and Test Procedures
Management, Engineering, and Assurance Reports Volume	GLAS Science Software Performance/Status Report
	GLAS Science Software Discrepancy Reports
	GLAS Science Software Engineering Change Proposal
	GLAS Science Software Test Report
[based on the NASA Software Documentation Standard - Software Engineering Program, NASA-STD-2100-91, July 19, 1991]	

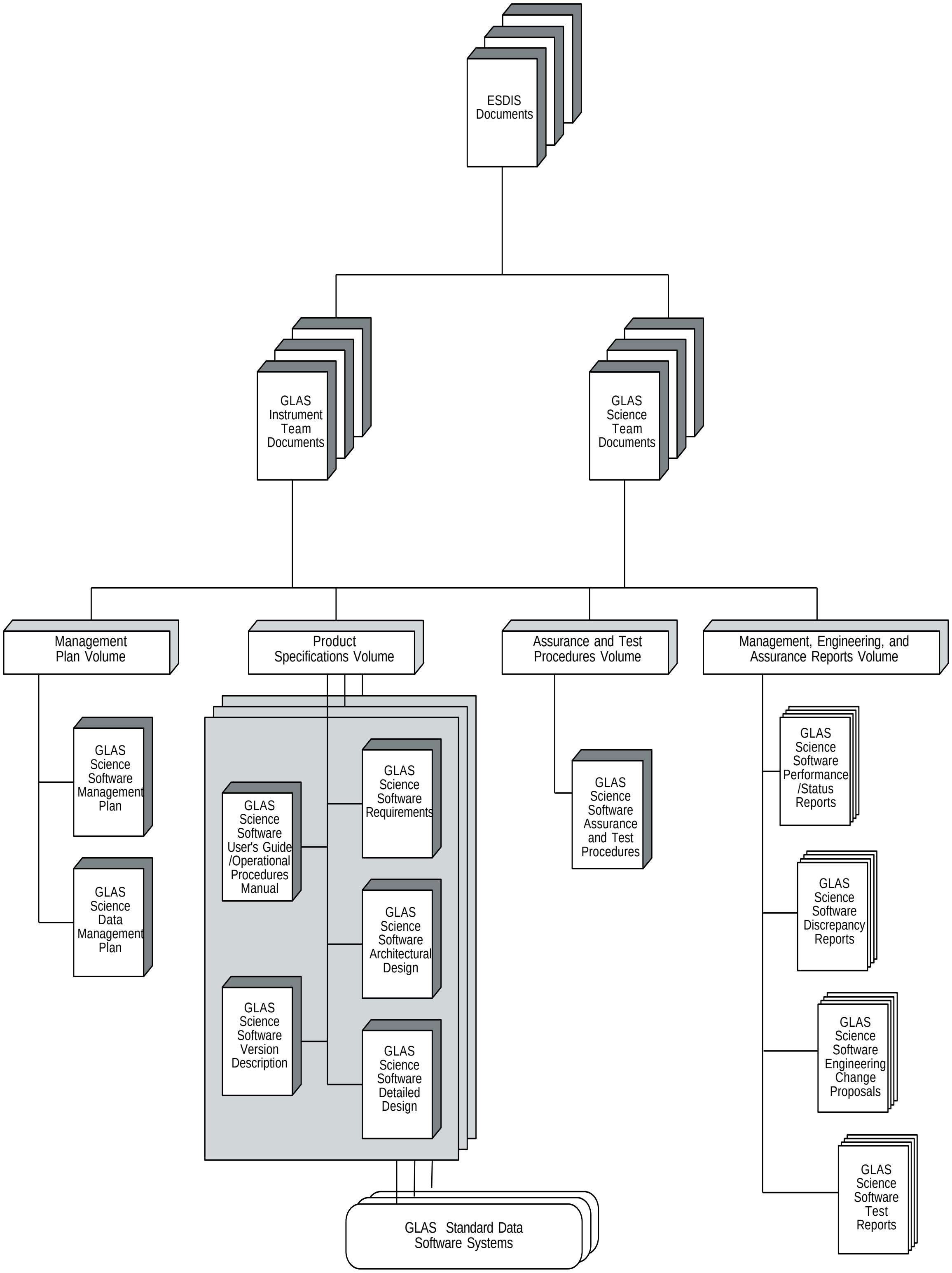

ESDIS Documents
GLAS Instrument Team Documents
GLAS Science Team Documents
Management Plan Volume
Product Specifications Volume
Assurance and Test Procedures Volume
Management, Engineering, and Assurance Reports Volume
GLAS Science Software Management Plan
GLAS Science Data Management Plan
GLAS Science Software User's Guide /Operational Procedures Manual
GLAS Science Software Version Description
GLAS Science Software Requirements
GLAS Science Software Architectural Design
GLAS Science Software Detailed Design
GLAS Science Software Assurance and Test Procedures
GLAS Science Software Performance /Status Reports
GLAS Science Software Discrepancy Reports
GLAS Science Software Engineering Change Proposals
GLAS Science Software Test Reports
GLAS Standard Data Software Systems

Figure 1-1 GLAS SDS Documentation Tree

Section 2

Related Documentation

2.1 Parent Documents

The GLAS Science Software Management Plan is a top level document. Within the context of the four-volume NASA software documentation standards [Reference 2.2c], this document represents the Management Plan Volume. It is not "rolled-out" from a parent document or volume.

The ESDIS and GLAS parent documents outside this software management plan are listed below.

a) *NASA Earth Observing System Geoscience Laser Altimeter System GLAS Science Requirements Document,* Version 2.01, October 1997, Center for Space Research, University of Texas at Austin.

2.2 Applicable Documents

The following documents are applicable to, or contain policies or references pertinent to the contents of this plan.

a) *Data Production Software, Data Management, and Flight Operations Working Agreement for GLAS*, TBD, NASA Goddard Space Flight Center

b) *NASA Software Documentation Standard Software Engineering Program*, NASA, July 29, 1991, NASA-STD-21000-91

c) *GLAS Science Management Plan*, December 31, 1995, Center for Space Research, University of Texas at Austin.

d) *GLAS Investigation Activities Plan*, December 31, 1995, Center for Space Research, University of Texas at Austin.

The highest level Data Item Description (DID) and section from the software documentation standards [Reference 2.2c, Appendix C] used to prepare the GLAS Science Software Management Plan document was NASA-DID-M000, the Management Plan DID.

2.3 Information Documents

The following documents provide background or supplemental information that may clarify or amplify material in this plan document.

a) *Data Production Software and SCF Standards and Guidelines*, January 1994, NASA Goddard Space Flight Center, 423-16-01

b) *Science Data Processing (SDP) Toolkit Requirements Specification for the ECS Project*, January 14, 1994, NASA Goddard Space Flight Center, 423-16-02

c) *Software Developer's Guide to Preparation, Delivery, Integration and Test with ECS*, Final, August 1995, Hughes Applied Information Technology Corporation, 205-CD-002-00

d) *SDP Toolkit Primer for the ECS Project*, December 19, 1995, Hughes Information Technology Corporation, 194-815-S14-00l

e) *A Science User's Guide to the EOSDIS Core System (ECS) Development Process*, February 1995, Hughes Applied Information Systems, 160-TP-003-001

f) *GLAS Science Computing Facility (SCF) Plan*, October 1997, NASA Goddard Space Flight Center, Wallops Flight Facility

g) *GLAS Science Data Management Plan*, October 1997, NASA Goddard Space Flight Center, Wallops Flight Facility

h) *Interface Responsibilities for Standard Product Generation using Science Investigator-led Processing Systems*, May 1998, NASA Goddard Space Flight Center, 423-42-03

Section 3

Purpose and Description of the GLAS Standard Data Software

The GLAS Standard Data Software consists of the software to do the standard data product and metadata generation and the software to support the functions of the GLAS Instrument Support Terminal (IST) and to perform instrument data trend analysis. The standard data product and metadata generation is called the ICESat Science Investigator-led Processing System (I-SIPS) Software. The software to support the IST functions is called the IST Software. The following sections present descriptions of the I-SIPS Software, the IST Software, and the roles and responsibilities of the GLAS Science, Instrument, and SDS Development Teams and the ESDIS. Figure 3-1 "GLAS SDS System Architecture" illustrates the required interactions between the GLAS SCF, the GLAS IST, and the EOSDIS in order for the SDS System to operate.

3.1 ICESat Science Investigator-led Processing System Software

The I-SIPS Software shall create the GLAS Level 1A, Level 1B, and Level 2 standard data products. The Level 1A, Level 1B, and Level 2 terms are defined in the Glossary. These products will be made available to the GLAS Science Team and to the science community. There are several planned deliveries of the software to the ICESat SCF; included in the delivery plan are at a minimum the ESDIS required Beta, V1, and V2 (final) deliveries. Details of these deliveries are discussed within this plan. The GLAS standard products are delivered to the DAAC for archival and dissemination to the science community. In addition, the I-SIPS Software will perform the metadata generation. The purpose of the metadata generation is to assess the standard data product generation performance, to assess the standard data product quality, and to provide descriptive information about the data products. The metadata will be delivered to the DAAC. The Science Team and ESDIS will have agreements to the format of the data products to be stored on the DAAC, the format and content of the metadata, and the method to deliver the products and metadata to the DAAC.

The I-SIPS Software will perform the following functions:

- create the Level 1A, Level 1B, and Level 2 GLAS standard data products at appropriate data rates and with sufficient precision to satisfy the requirements of the Science Team;
- assess the quality of the GLAS standard data products;
- assess the performance of the I-SIPS Software; and
- produce information describing the standard data products.

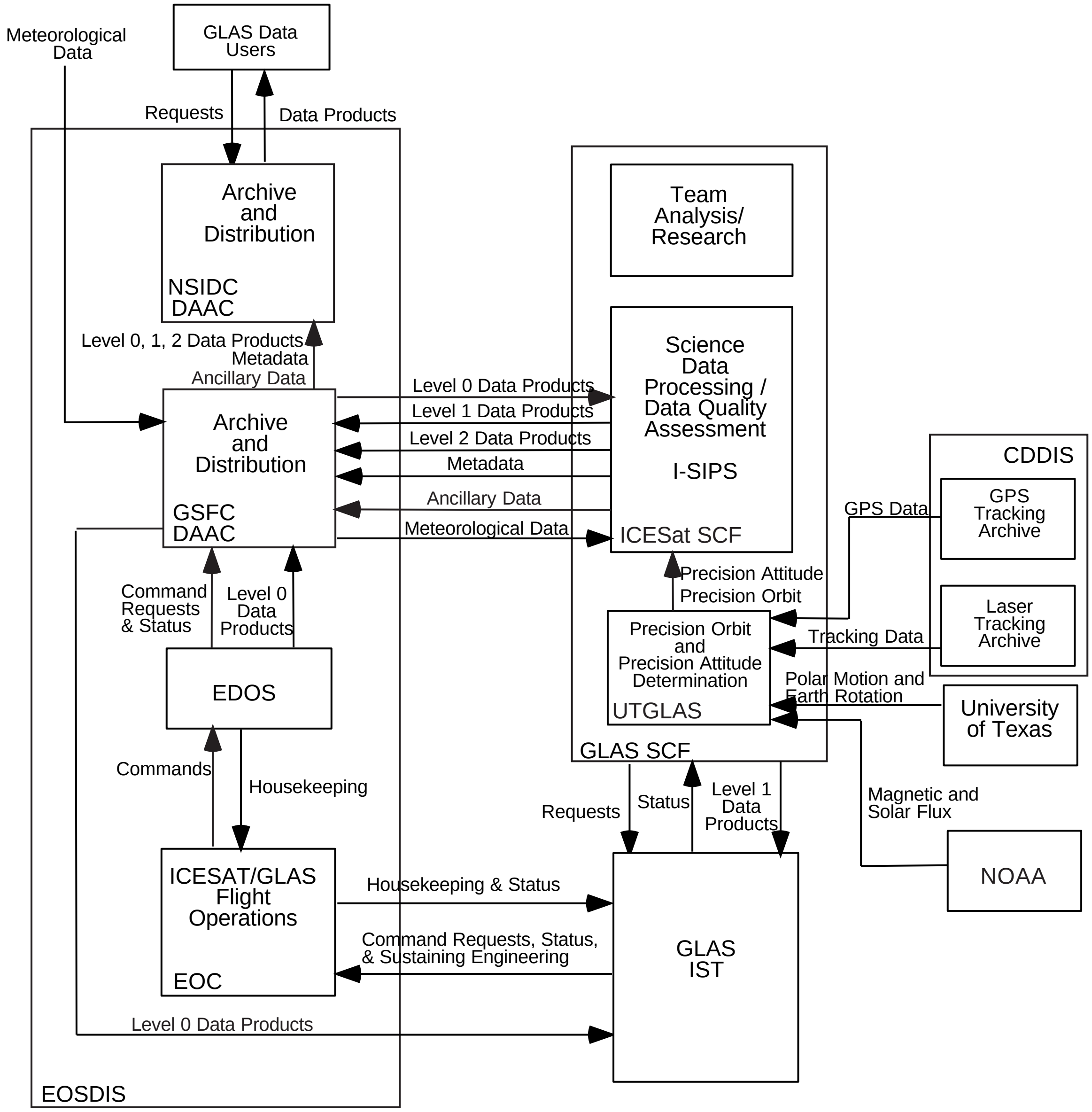


Figure 3-1 GLAS SDS System Architecture

The I-SIPS Software will accept as input: a) the GLAS intrument telemetry data stream from EDOS, b) GLAS standard data products, and c) ancillary data. The output primarily consists of the standard data products and the data product metadata.

Figure 3-2 "I-SIPS Top-Level Data Flow Diagram" on page 3-3 depicts the flow of data through the I-SIPS. Refer to the GLAS Product Specification Volume for detailed descriptions of the GLAS standard data products.

The I-SIPS Software will be designed and implemented under the direction of the Science Team to execute on the ICESat SCF node of the GLAS SCF. While the I-SIPS Software will be delivered to and operated on the ICESat SCF, the Earth, Science Data

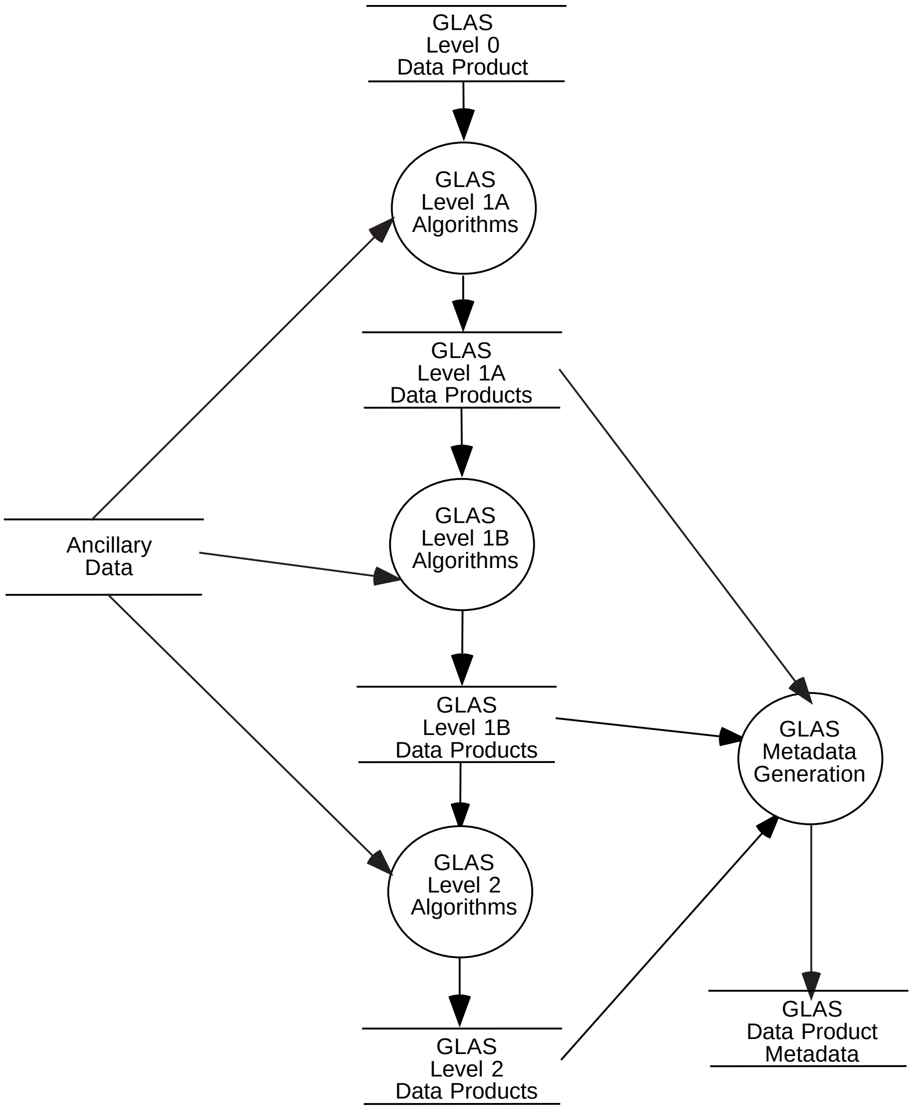


Figure 3-2 I-SIPS Top-Level Data Flow Diagram

and Information System (ESDIS) will oversee the I-SIPS activities. Additionally, the ESDIS requires that the software and its documentation is delivered to the DAAC for archive/backup.

3.2 GLAS Instrument Support Terminal Software

The GLAS IST Software includes the instrument command software and the instrument performance and health assessment software. The software interfaces with EOSDIS, and provides instrument command and monitoring capabilities to the Science Team and the Instrument Operations Team.

The instrument command software will support the preparation of laser altimeter operational command sequences and the validation of these command sequences prior to transmission and uploading to the instrument onboard the EOS ICESat spacecraft. The software will also support verification of command uplink and execution. The command software will be capable of supporting both real time and non-real time command operations. The command software will ensure that unauthorized or erroneous commands are not created and sent to the instrument or spacecraft. The IST software will support the modification of the flight software and the updating of flight software parameters.

The GLAS IST software will evaluate data received from both the EOS ICESat spacecraft and the GLAS instrument to determine the health and welfare of the laser and electronics. The data evaluated includes the output from the temperature, voltage, and current monitors incorporated into the flight hardware. The software will report data that exceed engineering threshold or limits values, and will raise flags identifying anomalous or erroneous instrument activity which may indicate aberrant sensor behavior or mission-threatening conditions. The instrument health assessment software will produce routine reports and graphical displays for the GLAS Science and Instrument Operations Teams to review and evaluate. The support GLAS IST Software will operate on the GLAS IST under control and responsibility of the Science and Instrument Operations Teams.

3.3 GLAS Team Roles and Responsibilities

3.3.1 Overview

The GLAS Science Team Leader is responsible for the development of the Standard Data Software and its delivery. This plan defines a SDS Development Team (SDT) to work under the Team Leader to support this effort. The Instrument Team will provide support to the GLAS IST Software development.

The Science Team Leader provides the official interface between the Science Team and the SDT. In addition to appropriate software development methodology, it is important to the development of successful Standard Data Software to have open lines of communication between the Science Team members and the SDT members. The SDT needs input from the Science Team regarding their data product and algorithm requirements, and the SDT needs to keep the Science Team aware of the steps being taken to assure the adequacy of the products. To facilitate this communication, the Science Team members will have the opportunity to review all of the SDT-produced documentation and to attend/participate in all the software reviews.

3.3.2 Science Team Leader Responsibilities

The GLAS Science Team Leader is responsible for all aspects of the GLAS SDS development. The Team Leader's specific responsibilities are contained in References 2.2c and 2.2d. Among this individual's responsibilities, the Science Team Leader will:

- Provide the interface between the Science Team and the SDT.
- Provide the SDT with Algorithm Theoretical Basis Documents (ATBDs) that define the required algorithms for processing the GLAS instrument data into the GLAS standard data products.

- Gain the Science Team's approval of the software development activities.
- Approve all software requirement and design documents.
- Attend each system design review and subsystem design reviews as needed.
- Provide directions or recommendations to the SDT as needed.
- Approve any system level change request and be a member of the software Change Control Board.
- Approve all system level acceptance test plans and reports.
- Participate in the test and analysis of software prior to each formal delivery to the ICESat SCF.
- Approve all formal releases of the SDS, and work with the Science Team to approve the software prior to delivery.
- Ensure that appropriate ESDIS requirements are satisfied.

3.3.3 Science Team Member Responsibilities

The Science Team members' inputs on the software design are crucial to the delivery of successful GLAS Standard Data Software. By attending design reviews and by reviewing design documents, the Science Team will influence the design of the GLAS SDS. Through the Science Team Leader, the Science Team will have approval for all software requirements. The specific responsibilities of each Science Team Member are defined in References 2.2c and 2.2d. During the GLAS SDS development, the Science Team responsibilities will include the following:

- Provide software requirements to the SDT through the ATBDs and other GLAS documents.
- Review Science Software requirements documents as produced by the SDT to determine if their specifications are complete and are correct.
- Review design documents which will be provided by the SDT. The Science Team will be invited to attend all system and subsystem design reviews.
- Supply or define test cases, code, and/or data sets.
- Participate in software acceptance and test data product validation, and participate in acceptance testing of each software version prior to delivery.
- Define/Produce any required ancillary data.

3.3.4 Instrument Team Responsibilities

The GLAS Instrument Team is responsible for providing support to the development of the GLAS IST Software. Among the team’s responsibilities are:

- Provide the SDT with information that define the processing of the GLAS instrument data into the instrument health monitoring products.
- Provide the SDT with requirements and information to develop the GLAS command generation software for the IST.

- Make directions or recommendations to the SDT as needed.

3.3.5 Standard Data Software Development Team Responsibilities

The GLAS SDS Development Team is responsible to the Science Team Leader for the development and delivery of the GLAS SDS defined in this plan. Among their responsibilities, the SDT will:

- Make regular status reports to the Science Team Leader.
- Inform the Science Team of the development activities and status through presentations or reports.
- Obtain from the Science Team, Instrument Team, and ESDIS the required information for proper software development.
- Provide the Science Team access to the software as it is ready for acceptance testing.
- Develop documentation to define all interfaces of GLAS with the ESDIS.
- Develop the software documentation for the GLAS SDS as required by ESDIS.
- Develop and deliver tested software with test cases to the ICESat SCF to facilitate acceptance testing.
- Develop and deliver tested software with test cases for the IST.
- Provide sustaining engineering for the GLAS SDS.

There will be a SDS Development Team Leader to ensure that these responsibilities are fulfilled.

3.4 ESDIS Responsibilities

The ESDIS is responsible for the following with respect to the SDS:

- Provide and maintain the Science Data Production (SDP) Toolkit and the Instrument Support Toolkit.
- Provide any requirements for standard data products and metadata, standard data product generation, and instrument flight operations support.
- Modification of the I-SIPS Software to execute on the DAAC, if desired.
- Archival of all I-SIPS Software source files, documentation, test information, and ancillary files.
- Provide the archival and distribution facility for the GLAS data products.
- Provide distribution of I-SIPS Software source files, documentation, information, and ancillary files.
- Provide user support for GLAS science data.

Section 4

Resources, Budgets, Schedules, and Organizations

4.1 Business Practices Definition and Revision Process

4.1.1 Definition of Activities

The following activities will be performed to determine budget and schedule compliance:

a) Tasks will be defined and assigned as activities in a Work Breakdown Structure (WBS).

b) Schedules will be determined based on the WBS.

c) Resources will be allocated to the activities in the WBS.

d) The schedule and planned resources will be monitored versus the actual status of activities and actual cost.

4.1.2 Method and Approach

The following procedures will be used to ascertain schedule and budget compliance by the Standard Data Software development effort:

a) Activities will be defined and a WBS will be created by the Standard Data Software Development Team (SDT) Leader and the SDT.

b) The SDT Leader will work with the SDT to develop a schedule for performing WBS task items, with an estimate of resources required. The schedule and resource estimation will be reviewed with the SDT Leader to resolve conflicts. The process will be iterated to attain an acceptable schedule.

c) The SDT will report progress toward goals, and alert the SDT Leader if tasks exceed or are expected to exceed projected schedules.

d) The SDT Leader will use commercial project management tools to track the WBS activities defined in Section 4.2.

e) If goals are missed or if tasks exceed or are expected to exceed projected schedules, the SDT Leader and the SDT will determine the cause and recommend a course of action.

f) Status and budget reviews will be regularly scheduled. Weekly status meetings will be held to discuss progress, problems and action items.

g) The SDT will provide a written performance/status report on each work area to the SDT Leader one day prior to the weekly status meeting. The reports are summaries that address work progress and any concerns.

4.1.3 Reporting, Monitoring and Revision

The SDT will prepare a Performance/Status Report to inform the SDT Leader of planned versus actual performance. The Performance/Status Report will include open action items, problems, newly identified risks, and status on the previously reported risks, along with recommendations for corrective action.

The SDT Leader will prepare technical progress reports for delivery to the GLAS Science Team Leader, providing summary/highlights/overview of the SDS development. The technical report will include, as a minimum: progress versus plans, future planned activities, and areas of concern.

The SDT Leader will use a project management tool to monitor software development progress versus schedule.

Revisions to completed and accepted software and its associated documentation will be made through the initiation and approval of an Engineering Change Proposal (ECP). The ECP process is defined in Section 10. Revisions which affect the tone or direction of software or documents in progress need the approval of the SDT Leader.

4.2 Work Breakdown Structure (WBS)

The following sections present the WBS activities and cost accounting methods. A top-level activity list for the GLAS SDS development effort is shown in Table 4-1 "GLAS Standard Data Software Work Breakdown Structure" on page 4-3. The GLAS SDS WBS and schedule will be placed within a commercial project management tool for use in software development effort planning, and for monitoring and tracking the delivery schedule and resource utilization

4.2.1 Activity Definition

The software development support is categorized by the software systems to be developed. The categories are further divided to reflect the GLAS SDS software development life cycle; the software development life cycle approach is defined in Section 6. As discussed in Section 6, each phase of the life cycle shall overlap somewhat; SDT Leader approval must be obtained prior to beginning any phase. At the top level, the GLAS SDS development effort is divided into three categories: 1) the Standard Data Software development, 2) the ICESat Science Investigator-led Processing System (I-SIPS) Software development, and 3) the GLAS Instrument Support Terminal (IST) Software development.

4.2.1.1 GLAS SDS Development (WBS 1.0)

This category includes those activities required in support of the GLAS SDS development.

4.2.1.1.1 GLAS SDS Requirements (WBS 1.1)

During the Requirements Phase, the SDT will perform the activities listed in Table 4-2. The requirements activity provides support for both the software development management planning element and the initial software development. The activities

Table 4-1 GLAS Standard Data Software Work Breakdown Structure

Category	Life Cycle Phase	Activity
1.0 GLAS SDS	1.1 Requirements	1.1.1 Define Software Development Management Practices 1.1.2 Determine SDS Requirements 1.1.3 Determine Input/Output Data 1.1.4 Support Science Team Algorithm Development 1.1.5 Prototyping
2.0 I-SIPS Software	2.1 Design	2.1.1 Prototyping 2.1.2 Develop I-SIPS Software Architectural Design 2.1.3 Develop I-SIPS Software Detailed Design 2.1.4 Develop User’s Guide/Operational Procedures Manual 2.1.5 Start Unit Development Folders
	2.2 Implementation and Testing	2.2.1 Develop Assurance and Test Procedures 2.2.2 Code Units and Test 2.2.2.1 Beta Delivery 2.2.2.2 Engineering (V1) Delivery 2.2.2.3 Launch (V2) Delivery 2.2.3 Integrate Units and Test 2.2.3.1 Beta Delivery 2.2.3.2 Engineering (V1) Delivery 2.2.3.3 Launch (V2) Delivery
	2.3 Acceptance and Delivery	2.3.1 Perform Acceptance Testing 2.3.1.1 Beta Delivery 2.3.1.2 Engineering (V1) Delivery 2.3.1.3 Launch (V2) Delivery 2.3.2 Develop Data User’s Handbook 2.3.3 Deliver Code and Support Installation and Testing 2.3.3.1 Beta Delivery 2.3.3.2 Engineering (V1) Delivery 2.3.3.3 Launch (V2) Delivery
	2.4 Sustaining Engineering and Operations	2.4.1 Maintenance Support 2.4.2 Operations Support
3.0 GLAS IST Software	3.1 Design	3.1.1 Design IST Software 3.1.3 Develop User’s Guide/Operational Procedures Manual
	3.2 Implementation and Testing	3.2.1 Develop Assurance and Test Procedures 3.2.2 Code and Test
	3.3 Acceptance and Delivery	3.3.1 Perform Acceptance Testing 3.3.2 Deliver Code and Support Installation and Testing
	3.4 Sustaining Engineering and Operations	3.4.1 Maintenance Support 3.4.2 Operations Support

of this phase will produce the following documents: 1) a Software Management Plan (this document), 2) a Standard Data Software Requirements Document and 3) a Data Management Plan. These documents are pertinent to the management and development of the Standard Data Software, and are to be completed prior to the completion of the Requirements Phase.

The SDT will support the development and assessment of the science algorithms to be used in the I-SIPS Software, as requested. Support will be provided to the Science Team for other documents and analysis related to the GLAS standard data product generation.

The software requirements document is a prerequisite for the software design phase of the I-SIPS Software and the IST Software. An edition of the requirements document approved by the SDT Leader shall precede the initiation of the design phase. Additional editions of this document may be released. The Science Team will review and approve the requirements prior to completion of the design phase. The final version of this document will be provided with the final delivery of the software.

4.2.1.2 I-SIPS Software Development (WBS 2.0)

The I-SIPS Software development involves the activities required to produce the software from the requirements generated in the WBS 1.1 activity. The I-SIPS Software was described in Section 3. The software development activities include the production of the associated documentation.

4.2.1.2.1 I-SIPS Software Design (WBS 2.1)

The first activity within the design phase is developing the architectural design and producing the software architectural design document. The architectural design document represents the first design delivery of the software product specification. Upon acceptance of an architectural design by the SDT Leader and with his approval, the detailed design activity will begin. During the detailed design activity, the Science Team will have the opportunity to review the architectural design and to provide additional information to the SDT. Prior to the completion of the detailed design, the Science Team will approve the architectural design. The software detailed design document will be produced during the detailed design activity and will be delivered with the software delivery package. During the detailed design activity, the Units Development Folder (UDF) contents will be defined and UDFs will be started for the detailed design units. The UDFs are intended to be a repository of design, implementation, and testing information for the I-SIPS Software and there is planned to be one UDF for each software unit.

With the approval of the SDT Leader, the implementation and testing phase will be initiated. The approval will be based on a completed edition of the detailed design document. Additional editions of the detailed design document may be released as necessary. The Science Team will review and approve the detailed design prior to the completion of the code implementation.

Other activities during the design phase are the development of the I-SIPS Software User's Guide/Operational Procedures Manual and the GLAS Science Software Data User's Handbook. The data user's handbook is a non-standard document designed more for the data user community than for the software product development process. These documents will be finalized during the implementation and testing phase, and will be provided with the final delivery of the software prior to launch.

During the design phase, the initial objectives and procedures for unit, integration, and acceptance testing will be developed.

4.2.1.2.2 I-SIPS Software Implementation and Testing (WBS 2.2)

The code implementation and integration will begin with the approval of the SDT Leader. Functional units of code will be implemented for a software product or algorithm. Units of code are defined as the smallest logical piece of a program. The unit must perform a testable function. The SDT will define the units during the detailed design phase. Units will be tested by the SDT members who coded them. Logical, operational groups of units will be integrated by the SDT to constitute a build. The build software product is subjected to internal software team integration testing. The integration testing will be performed by SDT members other than those who coded or integrated the units involved. A test data set will be developed to be included in the software delivery package. The final integration test step shall be to test the software product with the delivery test data set.

During implementation and testing of the units of code, the required information (as defined during the detailed design activity) is stored in the UDFs.

Upon the approval of the SDT Leader, integrated and tested software builds will transition to the acceptance and delivery phase. At a minimum, integration tested builds will be completed for a Beta (ß) Version, an Engineering (V1) Version, and a Launch (V2) Version.

Concurrent with the code development, unit testing, and code integration, the development of an assurance and test procedures documentation volume is completed. The software assurance and test procedures will be produced by the SDT, and approved by the SDT Leader prior to beginning integration testing events. The software assurance and test procedures are developed according to the guidelines set forth in Section 8 of this Plan. The appropriate sections of the software assurance and test procedures pertaining to acceptance testing of the I-SIPS Software shall be delivered with the final delivery of the software prior to launch.

The I-SIPS Software shall be developed and constructed in accordance with the specifications included in the Science team’s Algorithm Theoretical Basis documents (ATBDs).

4.2.1.2.3 I-SIPS Software Acceptance and Delivery (WBS 2.3)

The acceptance and delivery phase begins with the decision by the SDT Leader that a software product is ready for acceptance testing. The acceptance testing activity represents the application of testing to fulfill the software product assurance requirements in a near-operational environment. Acceptance testing will be performed by SDT members other than those who coded or integrated the units involved. Acceptance testing is performed in accordance with the assurance plan (Section 8 in this document) and the software assurance and test procedures. Acceptance testing and reporting will be performed on the Beta, Engineering (V1), and Launch (V2) Version deliveries; there may be additional acceptance testing and reporting on any SDT build, phased, or incremental code deliveries.

The software products, having passed acceptance testing on the ICESat SCF node and been approved by the SDT Leader, are delivered to the I-SIPS Team for installation in the production area of the ICESat SCF. The Science Team will approve the software delivery prior to its being placed in production mode on the ICESat SCF. The SDT will actively support the software installation, compilation, loading, and testing.

The final (V2 or launch) version of the software, its associated documentation, and test data constitute a required Project delivery and is subject to a formal delivery review. Included in the delivery is the fixed metadata which includes the GLAS mission description and high level data product descriptions. Upon ESDIS acceptance, the software products will be placed under formal Project change control. Any software and documentation changes will then be requested and approved through the Engineering Change Proposal (ECP) process.

4.2.1.2.4 I-SIPS Software Sustaining Engineering and Operations (WBS 2.4)

The sustaining engineering and operations phase is instituted upon the successful installation and acceptance of I-SIPS Software by the GLAS Science Team and the I-SIPS Team. Maintenance support is provided on an “on-call” basis for the I-SIPS Software. Should the I-SIPS Team detect a problem with the software system, a discrepancy report will be issued and, if required, an ECP will be submitted. The SDT will investigate, correct the problem, and report the software, documentation, and/or procedure changes. Upon request, the SDT will provide technical assistance to the I-SIPS Team. Further details of the sustaining engineering are provided in Section 7.

During ICESat mission operations, the I-SIPS Software will be executed on the ICESat SCF to produce the GLAS standard data products and their metadata. Further details of the operations activities are provided in Section 7.

4.2.1.3 GLAS Instrument Support Terminal (IST) Software Development (WBS 3.0)

The GLAS IST Software development will create the software products which support the GLAS instrument operations. These products will consist of programs to support instrument health monitoring and to facilitate instrument commanding. The development effort includes creating the documentation associated with the software products. The software will operate on the host processor workstations designated as the GLAS IST. The IST Software is developed under the direction of the Science Team with support from the Instrument Team. IST Software products will be subjected to both informal and formal reviews.

4.2.1.3.1 GLAS IST Software Design (WBS 3.1)

From the requirements developed in WBS 1.1, a design for the GLAS IST Software will be developed, and a design specification will be produced. Prior to the start of the software implementation phase, an edition of the design specification must be approved by the SDT Leader. The design specification will be included with the final delivery of the software. The Science Team will approve the design prior to the completion of software implementation.

Within the design phase, the required user documentation is developed to support the operation of the IST Software. The SDT will prepare a user's guide/operational procedures manual. This document will be delivered with the final delivery of the software.

During the design phase, the initial test plans for unit, integration, and acceptance testing will be developed.

4.2.1.3.2 GLAS IST Software Implementation and Testing (WBS 3.2)

The software implementation and testing phase will begin upon the approval of the SDT Leader. The units are defined by the SDT members who are doing the implementation. A unit will be tested by the individual who coded it.

Units will be integrated into expanding program segments for phased delivery. SDT members other than those who coded or integrated the units will perform the integration testing, to determine capability and design compliance. Acceptance test data will be produced with the support of the Instrument Team during this phase. The acceptance test data will be included with the delivery of the IST Software.

Sufficiently mature program integration sets, as determined by the SDT Leader, will transition to the acceptance and delivery phase.

During the software implementation and testing phase, the assurance and test procedures document for the IST Software will be developed in accordance with the guidelines set forth in the assurance plan section of this document (Section 8), with sufficient detail to manage the software testing and acceptance. This document will be approved by the SDT Leader prior to beginning integration testing and software product assurance. The portions of the assurance and test procedures document pertaining to acceptance testing will be provided with the final delivery of the software.

4.2.1.3.3 GLAS IST Software Acceptance and Delivery (WBS3.3)

The first activity in the software acceptance and delivery phase is to perform the software acceptance testing. The software will be tested with the acceptance test data, in compliance with the assurance and test procedures document. The software will be acceptance tested by SDT members other than those who coded the software. Upon verification of desired test results by the SDT and the SDT Leader, the IST Software will be submitted for Science Team acceptance review. Upon acceptance by the Science Team, the software will be installed on the IST by the Instrument Operations Team. The Instrument Operations Team will perform acceptance testing on the software using the acceptance test data and using the acceptance test procedures provided by the SDT. The SDT will provide training and on-site support.

Upon completion of program installation, acceptance testing, and training, the activities will transition to the sustaining engineering and operations phase.

4.2.1.3.4 GLAS IST Software Sustaining Engineering and Operations (WBS 3.4)

The SDT shall maintain the software as required during the software sustaining engineering and operations phase. Any software problems will be reported through either the discrepancy reporting process or the ECP process. The SDT shall investigate and correct any problems detected by the Science or Instrument Operations Teams. As required, the SDT will provide technical assistance to the Instrument Operations Team. Sustaining engineering and operations are further discussed in Section 7.

4.2.2 Cost Account Definition

At the top-level, all GLAS SDS cost accounts associated with the software development WBS are under an SDS fund account. The account management for the GLAS SDS activity is managed by the SDT Leader with the assistance of the GLAS SDS Resource Analyst. The GLAS SDS development activity is subordinate to the GLAS Science Team cost account. Within the NASA Goddard Space Flight Center (GSFC) cost accounts, direct labor, materials, and other indirect cost categories will be tracked by the SDT Leader and Resource Analyst.

The civil service staff labor and institutional charges will be identified separately from contractor support. Institutional charges applied to Project funds allocated to the GLAS SDS activity, including management fees and facilities charges, will be identified and included in appropriate government-mandated cost accounting items.

The labor costs incurred from the support services contractor organizations will be included in the software development activity cost accounts. The support service contractors shall supply copies of monthly invoices identifying levels of direct and indirect labor by category, associated labor total charges, applied overhead fees, general and accounting fees, and the corporate profit fee applied to the invoice. These invoices shall be tracked by the GLAS SDS Resource Analyst to monitor the rate of expenditure and the staffing level details of the support services contractors.

Other costs including supplies and materials will be recorded and tracked from NASA stock-stores and supply requisition orders, and from purchase requisition orders and delivery reports. The material costs will be monitored by the Resource Analyst to verify and report actual expenditures against planned expenditures.

The cost accounting shall be operated and maintained to provide sufficient detail to track and monitor actual costs and funds expenditures against the software development effort plan and funding schedule. Table 4-2 "Work Breakdown Structure Associated Cost Account Structure for the GLAS SDS Development" shows the GLAS SDS development effort cost accounts organization.

4.3 Resource Estimation and Allocation to WBS

This software development management plan section presents the resources available for support of the WBS activities. Resources addressed within the succeeding subsections include the schedules in Section 4.3.1; funding plans, sources, and bud-

Table 4-2 Work Breakdown Structure Associated Cost Account Structure for the GLAS SDS Development

ESDIS Fund Account	
Account	**Category**
GLAS SDS Costs	Civil Service
	Contract
	Materials and Supplies
	Other Costs

gets in Section 4.3.2; organization in Section 4.3.3, available and required support equipment in Section 4.3.4; and physical facilities, material, and other resource support in Section 4.3.5.

4.3.1 Schedules

Schedule preparation and presentation are handled from two aspects. First, the Project milestone chart for the GLAS spacecraft, the GLAS instrument development, and the GLAS science activities is used to schedule deliverables and required review events. Next, the WBS is used to fill in the SDS development schedule with the software development activities. The initial development of the SDS development schedule will use the Microsoft Project planning tool. This tool provides the basis for the preliminary top-level schedules, and readily supports the schedule refinement necessary as the software development effort and scope mature, and as more precise determinations of resources and work activities are obtained.

The project management tool and the master schedule are used to produce intermediate schedules for major activities and activity deliverables. These schedules are expanded subsets of the master schedule, and are used for specific planning and operational periods such as fiscal year 1998 or 1999. The intermediate schedules contain and reflect the dependence on the top-level milestones and activity transition points from the master schedule, but contain more detail than the master schedule. The SDS development schedule is included in Appendix A, WBS and Schedules. The schedule is presented in a timeline chart format.

The following information is to be contained in the schedule: milestones including ESDIS Project and GLAS deliveries, ESDIS Project and GLAS reviews and ancillary activities. Informal meetings, reviews, and progress assessment points are not required in the schedule, but may be included as necessary to support the planning process or in support of major or intermediate milestones such as deliveries.

Table 4-3 "Review/Delivery Provided to Standard Data Software Development Team" lists reviews and deliveries that will supply information to the SDT. The reviews and deliveries are required by the ESDIS Project. The dates for these reviews and deliveries will be included in the GLAS SDS schedule. As required, the SDT shall provide support for the reviews and deliveries.

Table 4-3 Review/Delivery Provided to Standard Data Software Development Team

Review/Delivery	Provider
Instrument System Requirements Review (SRR)	Instrument /Science Team
Conceptual Design and Cost Review (CDCR)	Instrument /Science Team
Instrument Preliminary Design Review (PDR)	Instrument /Science Team
Instrument Critical Design Review (CDR)	Instrument /Science Team
Operations Readiness Review	Instrument /Science Team
Delivery of Final Calibration Plan	Instrument Team
Delivery of Final Scientific Data Validation Plan	Science Team
Delivery of Final ATBD	Science Team

Table 4-4 "Review/Deliveries Supported/Provided by Standard Data Software Development Team" lists reviews and deliveries to be supported or provided by the SDT during the software development. The dates for these reviews and deliveries will be included in the schedule.

Table 4-4 Review/Deliveries Supported/Provided by Standard Data Software Development Team

Review/Delivery	Required by	Description
Science System Requirements Review (SRR)	ESDIS Project	formal review, SDT will support
Science Preliminary Design Review (PDR)	ESDIS Project	formal review, SDT will support
Science Critical Design Review (CDR)	ESDIS Project	formal review, SDT will support
Beta Version Software Delivery Readiness Review	ESDIS Project	formal review, SDT will support
Beta Version Software Release	ESDIS Project	formal software product package delivery, SDT will provide the delivery
Engineering Version (V1) Software Delivery Readiness Review	ESDIS Project	formal review, SDT will support
Engineering Version (V1) Software Release	ESDIS Project	formal software product package delivery, SDT will provide the delivery
Launch Version (V2) Software Delivery Readiness Review	ESDIS Project	formal review, SDT will support

Table 4-4 Review/Deliveries Supported/Provided by Standard Data Software Development Team (Continued)

Review/Delivery	Required by	Description
Launch Version (V2) Software Release	ESDIS Project	formal software product package delivery; the final software product specification document set is required with this delivery, SDT will provide the delivery
Delivery of Final Science Data Management Plan (SDMP)	ESDIS Project	formal document, SDT will provide this document
Delivery of Final Science Software Management Plan (SSMP)	ESDIS Project	formal document, SDT will provide this document
Delivery of Final Software Product Specification Document	ESDIS Project	formal document, SDT will provide this document
Delivery of Final Science Computing Facility Plan	ESDIS Project	formal document, the SDT provides support
Delivery of Software Requirements Specification Document	SDT	roll-out of the software product specification document, SDT will provide this document
Delivery of Software Architectural Design Specification Document	SDT	roll-out of the software product specification document, SDT will provide this document
Delivery of Software Detailed Design Document	SDT	roll-out of the software product specification document, SDT will provide this document
Delivery of Software User's Guide/ Operational Procedures Manual	SDT	roll-outs of the software product specification document, combined into one document, SDT will provide this document
Delivery of Data User's Handbook	SDT	data description document for GLAS data products, SDT will provide this document
Delivery of Software Assurance and Test Procedures Document	SDT	contains assurance and test objectives and procedures, SDT will provide this document

Table 4-5 "Standard Data Software Development Team Required Activities" lists the activities required by the SDT that will be accommodated in the schedule. The schedule will at a minimum reflect the duration of the activity.

4.3.2 Funds and Budgets

Funds and budget information will be included in Appendix B when appropriate.

Table 4-5 Standard Data Software Development Team Required Activities

Activity	Description
Requirements	software product conception and initiation, requirements analysis
Design	architectural and detailed design analysis and development
Implementation and Testing	software implementation, integration, and integration testing
Acceptance and Delivery	software acceptance testing, product delivery and support
Sustaining Engineering and Operations	software product maintenance and operational support

4.3.3 Organization

This section describes the organizational structure for carrying out the management activities and processes associated with GLAS SDS development. To establish the basis of control and authority, the organizational chart for the GLAS SDS development is depicted in Figure 4-1 "GLAS Standard Data Software Development Organization" The key person for this effort is the SDT Leader. The SDT Leader oversees the development of the two SDS systems: 1) the I-SIPS Software, and 2) the IST Software. He directly communicates with and is responsive to the GLAS Science Team Leader. In addition, he works closely with the GLAS Instrument Team so as to develop appropriate software for the IST. As indicated by the dotted line in the organizational

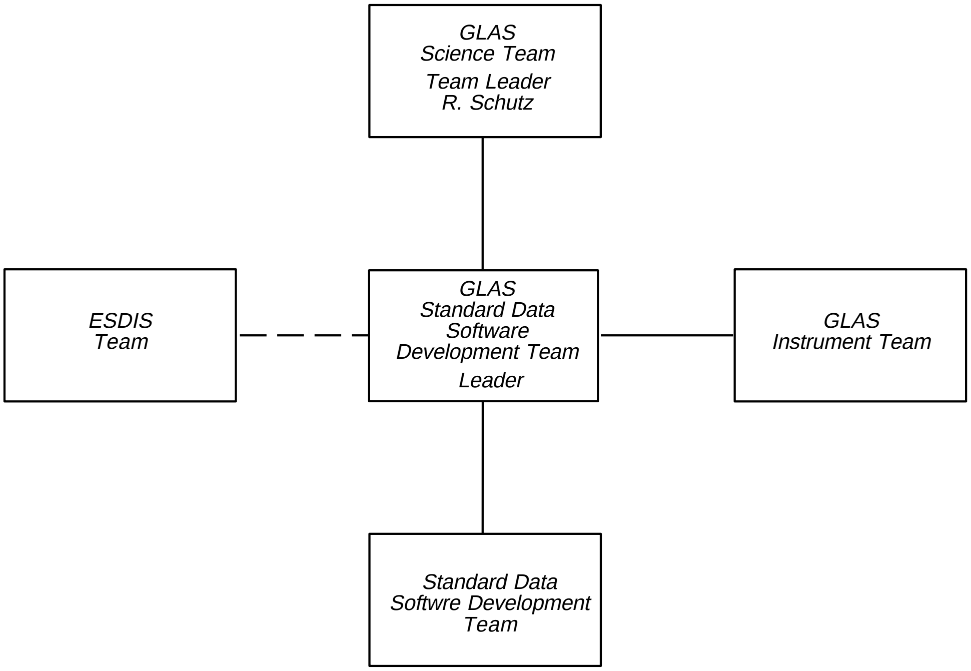


Figure 4-1 GLAS Standard Data Software Development Organization

chart, he will remain cognizant of, and be responsive to, applicable software documentation and development guidelines established by the ESDIS Team.

4.3.4 Equipment

This plan section provides a top-level description of the equipment required in the development of the GLAS SDS. All computer and support equipment for the GLAS SDS development effort will be subject to NASA GSFC standard property management procedures.

4.3.4.1 GLAS Science Computing Facility Equipment

This category includes the computer systems and associated support equipment identified as the GLAS Science Computing Facility. The GLAS SCF is composed of several nodes, one of which is the ICESat SCF node where the I-SIPS Software is developed and executed. The specific processors, peripherals, and associated support equipment will be identified in the *GLAS Science Computing Facility Plan*, Information Document 2.2f, a Project-required GLAS Science Team document.

4.3.4.2 GLAS Instrument Support Terminal Facility Equipment

The IST facility equipment category includes the computer workstation(s) and associated support equipment identified as the GLAS Instrument Support Terminal. The requirements and specifications for this equipment category will be determined by the GLAS Science Team. The function of this facility will be to support GLAS instrument operations such as command, instrument health and welfare monitoring, and instrument performance assessment. In the instrument operations software development process, it will provide the platform for code development, integration, testing, and acceptance. This facility equipment should support access to the UNIX system and tools by the instrument team and by the software development team, and shall support required operational instrument data access via standard network connections. The IST shall be operated in a secure manner to prevent unauthorized use.

4.3.4.3 Personal Computer Equipment

The personal computer equipment category consists of personal computer systems required in support of the GLAS SDS development and operation. Personal computers will consist of Apple Macintosh computer systems and their peripherals, and DOS/Windows-based computer systems and their peripherals will also be used. The personal computers shall be capable of supporting the planned documentation, project management, and network support tools and software. These personal computers shall be capable of providing network connectivity to computer and workstation equipment.

4.3.5 Materials, Facilities, and Other Resources

The GLAS SDS development facilities will be located at Wallops Island, Virginia, at Greenbelt, Maryland, and at off-site locations. All materials and supplies associated with the on-site GLAS SDS development effort will be obtained from GSFC stocks/stores or purchase requisitions. All on-site facilities will be occupied and used by Civil Service and approved contractor support personnel.

The long lead procurement items and critical resources for the GLAS Project will be covered under the GLAS Science Computing Facility Plan.

4.3.6 Management Reserves

There are no specified reserve funds associated with the GLAS SDS development effort.

4.4 Work Authorization

The majority of the SDS development work will be performed by a contractor through a NASA/GSFC support services contract. Work will be assigned to a task within the contract. The SDT Leader will provide to the contract a statement of work describing the task. The contractor organization will respond to the statement of work with planned activities and corresponding manpower estimates.

In addition, civil servants will be authorized to work on the SDS development task. The contractor may also subcontract out some the task work.

Work within the task will be authorized through the use of action items. Action items will be created by the SDT Leader or other authorized SDT members. The action items will be assigned to an SDT member (civil servant, contractor or subcontractor) who will determine what needs to be done to complete the action item and estimates the amount of time to complete. The SDT member can divide the action item work among other SDT members. The action items will have due dates and will be tracked to ensure work is progressing and completed on time. The SDT has developed a system for posting, tracking, and reporting action items.

Section 5

Acquisition Activities Plan

This Acquisition Activities Plan provides a definition of the activities that will be undertaken to acquire the GLAS Standard Data Software (SDS), and specifies management and assurance requirements. This plan covers all aspects of the life cycle for the software, including procurement.

5.1 Procurement Activities Plan

This section describes the procurement activities and events to be conducted, and identifies who will be responsible for these activities and events.

5.1.1 Procurement Package Preparation

The GLAS SDS will primarily be developed by NASA support services contractor organizations. The SDS Development Team (SDT) Leader will procure software development services through tasks under a GSFC support services contract(s).

Procurement packages for GLAS SDS development tasks will include: an appropriate Statement of Work with a reference to this document for the WBS and its descriptions, specifications for data requirements, and the schedule. The SDT Leader will develop associated schedule and cost information.

5.1.2 Proposal Evaluation

The GLAS proposals will be evaluated as part of the overall evaluation for the awarding of the GSFC support services contract(s).

5.1.3 Contract Negotiation

The support services contracts will be negotiated and awarded on schedule as they support multiple programs at GSFC. If there is a delay in the awarding of the new contract, the existing contract will be extended until the awarding.

5.1.4 Procurement Risks

The SDT Leader will identify and describe procurement risks and contingencies in terms of schedules and budgets. An inherent risk of using support contractors is that schedules and budgets may be impacted if a contract is re-bid during the software development process, and a new contractor is selected. The general procedure for incoming contracting companies, however, is to offer employment to the incumbent personnel, and have the personnel continue in their present duties.

5.2 Organizational Requirements and Life Cycle Adaptations

5.2.1 Business Practices, Resources, and Organizational Requirements

The SDT Leader will be the Technical Monitor for the GLAS SDS development task of the GSFC support services contract(s). The SDT Leader will use a project management software tool to monitor planned contractor performance versus actual schedules and costs, and will report status to the GLAS Science Team Leader.

The SDT Leader will evaluate task proposals from the support contractors for: appropriateness of schedule and costs, the contractor's understanding of the work to be accomplished, the contractor's capability to perform the work, the availability of qualified contractor personnel and physical resources, and the standards and practices to be followed. The SDT Leader will negotiate with the GLAS SDS task contractor representative(s) to resolve matters of costs, schedule, products and deliverables.

5.2.2 Life Cycle Adaptations and Approved Waivers

The contractors will develop, test, implement and maintain the GLAS SDS that is contracted per the life cycle described in Section 6.0 of this document. A waiver to permit a deviation from the defined GLAS software development life cycle process may be approved by the SDT Leader, but only if the software developer makes a compelling written case for doing so.

5.3 Management Approach

5.3.1 Software Management Responsibilities

The SDT Leader has the ultimate accountability for the success of the software development process. The SDT Leader will delegate logical functional groupings of software development tasks within the contractor team(s).

As Task Monitor, the SDT Leader will develop statements-of-work, budgets and schedules, and will monitor contractors performance. The contractor team(s) will provide the SDT Leader with monthly progress reports, monthly financial status reports, and interim informal status reports.

5.3.2 Categorization and Classification Policy

The smallest logical and testable entity in the GLAS SDS development process is the unit, defined as the code that implements a testable aspect of the requirements.

An integrated set of units is identified as a build. Each new build adds one or more units to the capability previously implemented and tested. The build concept minimizes risk and optimizes performance through incremental development.

Each build will undergo verification and validation, the process which ensures that the software satisfies functional requirements and that the software yields the right products.

5.3.3 Management Mechanisms

This section describes how the GLAS SDS development activity will function, from the management point of view.

5.3.3.1 Requirements Development and Control

Requirements applicable to the GLAS SDS will come from two organizational divisions. The first organizational source for requirements is the GLAS investigation. The GLAS investigation includes the elements such as the Instrument Team and the Science Team. Another source for requirements is the ESDIS Project.

The individual responsible for coordinating and controlling these requirements is the SDT Leader.

5.3.3.2 Schedule Development and Control

Software scheduling levels will follow the WBS as defined in Section 4.2 and illustrated in Table 4-1 "GLAS Standard Data Software Work Breakdown Structure". The SDT Leader will be responsible for assuring that the SDS elements are integrated with the Science Team and Instrument Team schedules.

A Gantt Chart, a graphic view of the WBS schedule, will be used to track the progress of the WBS tasks. A Tracking Gantt Chart will illustrate the differences in the planned schedules and the actual schedules to alert the team to these differences. The SDT will report progress toward goals, and alert the SDT Leader if tasks exceed or are expected to exceed projected schedules. The SDT will prepare Performance/Status Reports which will inform the SDT Leader of significant variances in planned versus actual performance. The Performance/Status Report will include progress to date, open action items, problems, newly identified risks, and status on the previously reported risks, along with recommendations for corrective action.

If goals are missed, or if tasks exceed or are expected to exceed projected schedules, the SDT Leader with input from the SDT will determine the cause and recommend a course of action.

5.3.3.3 Resource Development and Control

The SDT Leader will work with the SDT to develop a schedule for performing WBS task items with an estimate of resources required. The schedule and resource estimation will be reviewed with the SDT Leader to resolve conflicts. The process will be iterated to attain an acceptable schedule and resource estimate.

Regular status meetings will be held to discuss progress, problems and action items. The Performance/Status Reports (produced by the SDT on each work-area) will provide the SDT Leader with information regarding progress and any problems with task completion. The reports will be provided one day prior to the status meeting.

5.3.3.4 Internal Review Control

The SDT Leader will conduct technical reviews for each major data production software component scheduled at appropriate points within the software life cycle defined in Section 6.1.1. Different major software components may have individual reviews, but related components will be reviewed at the same time.

5.3.3.5 External Review Concepts

As required, the SDT Leader will support ESDIS initiated meetings, to discuss and to resolve any technical and programmatic issues related to the SDS, the GLAS SCF, and the GLAS IST. The SDT Leader will support Science and Instrument Team reviews, when invited.

Known external reviews are included in the WBS schedule.

The content of these reviews is defined in Section 6.3.1 but will include, as required, information describing the following:

a) progress since the last review;

b) activities planned for the next review period;

c) short and long term schedules;

d) any proposed changes to standard data products and/or input data;

e) current estimates of processing and storage for standard data products;

f) team organization changes;

g) identified risks and plans for their mitigation;

h) issues and concerns; and

i) budget allocation to work areas versus actual expenditures in these areas.

5.3.3.6 Board Support

The SDT will provide support, as requested, to the GLAS SDS and ESDIS Change Control Boards. This support will include, but not be limited to, Engineering Change Proposals (ECP) evaluations, presentations, and presentation materials.

5.3.3.7 Management and Control

Management and control of the design process are described above in Section 5.3.3.2 and in Section 6.1. Baselining of the products is described in Section 6.2.1.

5.3.4 Documentation Requirements

The SDT will produce documents during each phase of the software development life cycle. These documents are discussed in Section 6. The documentation requirements are approved and controlled by the SDT Leader.

5.3.5 Risk Management

The Risk Management Plan identifies the areas of risk and the methods to be employed to mitigate these risks. The Risk Management Plan is contained in Section 9.

5.3.6 Configuration Management

The GLAS SDS configuration management requirements are:

- ensure that the software system configuration is identifiable and well documented.
- ensure that changes and/or additions to the software reflect a thorough consideration of all system elements and interfaces.
- provide for controlled changes to the software and its supporting documentation.
- provide the mechanism necessary for smooth continuity of service in an environment subject to contractual change every five years.
- provide for controlled storage of and access to the software baseline.

The configuration management techniques and activities to be employed by the SDT are defined in Section 10.

5.3.7 System Assurance and Integration

GLAS SDS quality assurance activities and testing are addressed in the Assurance Plan, Section 8. Software integration is discussed in Section 6.

5.3.8 Deviation and Waiver Procedure

Any deviations or waivers from the processes outlined or defined in the baselined management plans, requirements documents, or design documents will be documented and approved through the ECP process. The deviation/waiver must be fully described, the consequences of not accepting the deviation/waiver must be explained, and approval must be obtained from the SDT Leader or the GLAS SDS Change Control Board. Deviations/waivers that do not affect any external interfaces must have the approval of the SDT Leader. Deviations/waivers that do affect any external interfaces must have the approval of the GLAS SDS Change Control Board.

Any deviations or waivers from the processes outlined or defined in management plans, requirements documents, or design documents that are not yet baselined must be approved by the SDT Leader.

5.3.9 Maintenance of Management Plan

This Plan will be updated periodically. The final release is considered the baseline and is placed under configuration management. The baselined Plan is updated through the ECP process; updates to the baselined Plan must be approved by the GLAS SDS Change Control Board.

5.4 Technical Approach

The pertinent schedule information for each of the processes described in this section is contained in Section 4.3.1.

5.4.1 System Requirements and Constraints

The functional requirements of the SDS are:

- Create the Level 1 and Level 2 GLAS standard data products.
- Create the GLAS standard data product metadata.
- Support the GLAS instrument operations.

These functional requirements were discussed in more detail in Section 3.

The I-SIPS Software portion of the GLAS SDS will execute on the ICESat SCF and the IST Software portion will execute on the GLAS IST. The GLAS SDS will be developed on both these systems. The ICESat SCF is described in Information Document 2.3f; the GLAS IST is described in TBD.

The SDT has chosen Fortran 90 to be the main programming language for the SDS.

The IST Software is constrained to use the Instrument Support Toolkit provided by the ESDIS Project.

The SDS shall adhere to ESDIS requirements when retrieving data from or delivering data to the Project facilities.

The implemented I-SIPS Software will conform to ESDIS coding standards to enhance portability of the software and to remove dependencies on any one hardware/software system/vendor.

Prototyping will be used by the SDT during the requirements definition and the design specification when it enhances these processes. See Section 6.1.2 for a discussion of prototyping.

5.4.2 Integrated System Description

The GLAS SDS system includes the I-SIPS Software and the IST Software. The GLAS SDS is described in Section 3. The I-SIPS Software will be delivered to the ICESat SCF node of the GLAS SCF and will be operated by the I-SIPS Team. The IST Software will be delivered to the IST and operated by the Instrument Team. An interface allowing the passage of data and information between the DAAC environment and the SCF environment will exist.

5.4.3 Software Requirements Definition Process

The GLAS SDS requirements will be defined during the requirements phase of the software development. The requirements phase is defined in Section 6.

5.4.4 Software Design and Implementation Process

The software design will occur during the software design phase of the GLAS SDS development. The implementation process will occur during the software implementation and testing phase. The phases are defined in Section 6.

5.4.5 Software Test and Delivery Process

The planned GLAS SDS testing and assurance activities are described in Sections 6 and 8. Software unit, build, and integration testing will occur during the software implementation and testing phase of the software development. Acceptance testing will occur during the software acceptance and delivery phase. The delivery and installation processes and requirements are discussed in Sections 6 and 11.

5.4.6 Software Maintenance and Updating Process

The SDT will provide maintenance support to the GLAS SDS according to the plan described in Section 7. Delivered software and documentation updates will be initiated through an ECP; the update process is discussed in Section 7. Delivered software and documentation updates will be implemented following change control guidelines as defined in Section 10.

5.4.7 Software System Engineering

The GLAS SDS development will follow a modified life cycle approach to software system engineering. The life cycle approach employed by the SDT is discussed in Section 6.

5.4.7.1 Implementation Policies and Standards

The policies and standards imposed by the Science Team and the ESDIS Project will be used.

5.4.7.2 Interface Control Process

The interface control process for the GLAS SDS is defined in Section 6.4.

5.4.7.3 Data Generation and Management Process

The GLAS Science Data Management Plan will define external data required for the generation of the GLAS Standard data products.

5.4.7.4 Performance Assessment Process

To assess the performance of the GLAS SDS while it is being developed, quality assurance and verification and validation (V&V) activities, as discussed in Section 8, will be performed; the progress of the development will be tracked against the schedule, and formal ESDIS Project reviews will be supported. The quality assurance and V&V activities will provide a measure of how accurately the software meets the requirements and the design, and how well it performs. The results of the V&V activities will provide a record of test pass/fails which will be a measure of how well the software is implemented. Reviews and schedule monitoring will provide data on the status of the software.

5.4.7.5 Operations Maintenance Process

The operations for the I-SIPS Software will be performed by the I-SIPS Team. The operations for the GLAS IST will be performed by the Instrument Operations Team.

The sustaining engineering for all GLAS SDS will be performed by the SDT. The operations and sustaining engineering activities are defined in Section 7.

Section 6

Development Activities Plan

6.1 Methodology and Approach

This section of the Science Software Management Plan addresses the GLAS Standard Data Software (SDS) development methodology, in terms of the development engineering approach, the integration and delivery guidelines, and the engineering support environment.

6.1.1 Development Engineering

All software development for the GLAS SDS will follow a well-defined software life cycle plan with adequate documentation generated and reviews held. The approach to be taken, following the guidelines of the NASA Software Engineering Program [Applicable Document reference 2.2c], will be to define and document requirements thoroughly before beginning design, and to use prototyping to refine requirements, verify critical areas of the design, and mitigate any higher risk elements. The SDS Development Team has chosen this approach to ensure that reliable, efficient, portable, maintainable, and reusable software is developed. The software will be built as software units; then, the units will be grouped and released in phases, identified as deliveries or builds, each having increasing capabilities.

The SDT will document the team's programming standards and guidelines for the SDS. All developed software shall be in compliance with the standards developed by the SDT. These standards, guidelines, and recommendations will increase the reliability, maintainability, portability, reusability, operability, and efficiency of the developed I-SIPS and IST software. Additionally, the software development engineering approach shall accommodate sufficient data and process error handling for error detection, isolation, and handling in order to ensure maintainability, efficiency, and reliability.

6.1.1.1 Life Cycle Management

The software life cycle described herein will be applied to all software management levels within the GLAS SDS development, and each phase will be addressed for all SDS, whether internally developed, acquired, or adapted.

The phases of the GLAS SDS development life cycle and their major activities are listed below:

- Requirements Phase
  - Concept and Initiation
  - Requirements Development
  - Prototyping
- Design Phase

  - Prototyping
  - Architectural Design
  - Detailed Design
- Implementation and Testing Phase
  - Implementation
  - Integration and Test
- Acceptance and Delivery Phase
  - Acceptance
  - Delivery and Installation
- Sustaining Engineering and Operations Phase
  - Operations
  - Maintenance

Although the life cycle is presented as a serial process, the life cycle approach is an iterative process with a re-visitation of earlier phases and activities as may be required to refine requirements, specifications, interfaces, and procedures. Repetitions of the life cycle activities will be used to accommodate the incremental deliveries of the software products. The software will be completed in builds, which are stand-alone groupings of software modules satisfying a portion of the required software functions, and able to operate independently. Each successive build will incorporate more of the required capabilities of the completed software. Therefore, there will be several iterations through some phases. Each phase of the life cycle will end with an informal review, focusing on the required documentation, which will serve as a maturity check before proceeding to the next life cycle phase. The SDT leader will approve procession to the next life cycle phase. The GLAS Science Team will formally approve each phase prior to the completion of the next phase. Phases may also overlap since the software is modular, with some modules progressing to completion faster than others. The sections which follow describe the phases of the life cycle and the documentation which will be generated during each phase.

The software life cycle phases and subphases are represented in Figure 6-1 "GLAS SDS Development Evolutionary Life Cycle Model" on page 6-3. This diagram shows the NASA standards waterfall process as adapted for the development of the GLAS SDS. Included in the figure are the formal reviews to which the SDS will be subjected. The product deliverables for each life cycle phase and subphase are shown in Table 6-1 "Software Engineering Master Schedule" on page 6-8. The GLAS SDS life cycle approach correlates with the activities in the work breakdown structure and schedule discussed in Section 4.

Throughout the life cycle process, the Performance/Status Report will be used to document development activities. The Discrepancy Report will be utilized to document any reported problems, failures, or deficiencies. The Engineering Change Proposal will be applied to identify, evaluate, authorize, and perform any changes,

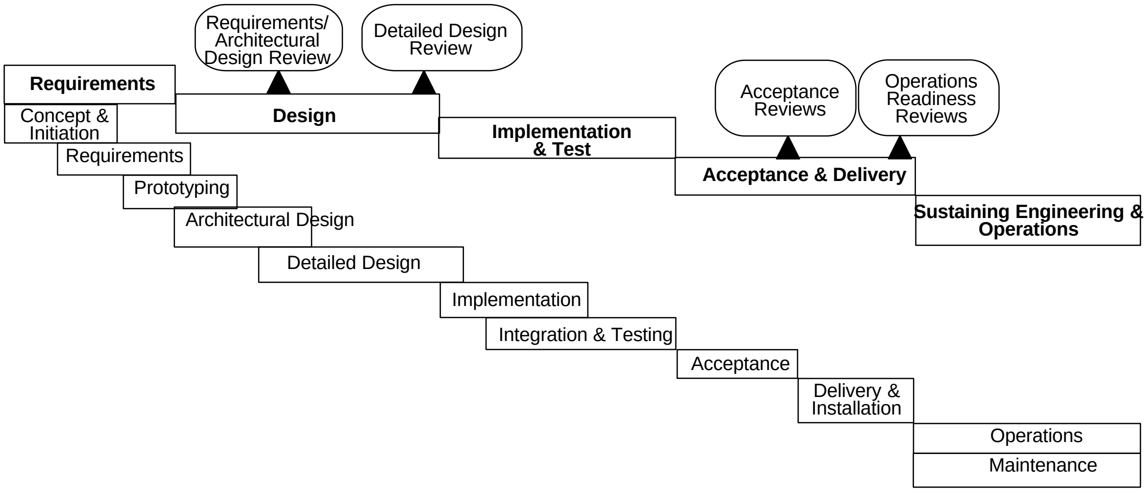


Figure 6-1 GLAS SDS Development Evolutionary Life Cycle Model

modifications, or revisions required to be made to either documents or software products above the baselined versions.

6.1.1.1.1 Requirements Phase

At the beginning of the Requirements Phase, it is determined what software is to be produced and for whom it is to be produced. The practices for managing the development of the software are also defined. Software concept and initiation is the title given to these processes. At the beginning of the Requirements Phase, during software concept and initiation, the following documents will be produced:

- GLAS Science Software Management Plan
- GLAS Science Data Management Plan

Because the GLAS SDS development is a small project (fewer than 20 developers) and the effort will be managed by one person, the SDT Leader, a single Science Software Management Plan, with expanded sections to cover software development practices, will provide the management practices definition. This phase transitions to the requirements development through the approval of the SDT Leader.

During the software requirements development the SDS, its products, and its operating environment are analyzed to determine the requirements on the software systems. The GLAS Science and Instrument Teams are interviewed for additional requirements. Constraints on the software are determined. The software requirements development will produce a software requirements document for the I-SIPS and IST Software. The software requirements document specifies the functional, performance, and interface requirements of the software. It also specifies the major characteristics of and the constraints on the software. Additionally, the software requirements document lists the design goals and discusses the partitioning of the requirements for phased delivery of the software. During the Requirements Phase, it may become necessary to perform some prototyping to determine the details of the requirements and/or their feasibility.

With the approval of the SDT Leader, the Requirements Phase transitions to the Design Phase. As necessary, the Requirements Phase or portions of it will be ongoing until all requirements are identified and documented. The requirements will be formally presented to the Science Team at the Requirements/Architectural Design Review.

6.1.1.1.2 Design Phase

During the software Architectural Design activity of the Design Phase, the SDT will define the logical/functional design of the GLAS SDS architecture to at least one level of decomposition. The architectural design documentation will describe the design rationale, the data relationships, and the external interfaces; the allocation of the software requirements to the architectural design elements will be defined. The documents will contain the data flow diagrams with supporting descriptions.

The architectural design will be presented to the Science Team and the SDT at the Requirements/Architectural Design Review.

With approval of the SDT Leader, the detailed design process begins. The detailed design will define the complete design of the software. The decomposition of the software will be defined to the unit level. The design of all interfaces and the mapping of the architectural design to the detailed design will be included.

The Unit Development Folder contents will be defined during the detailed design activity and UDFs for each detailed design unit will be started. The UDFs are completed during the implementation and testing phase.

The detailed design document will contain module-by-module sections, each composed of a processing narrative, interface description, a program design description, and data organization. The detailed design document will include enough detail to enable a programmer to write the software code to implement the design. The detailed design document for the I-SIPS Software will incorporate the science algorithms specified in the GLAS Algorithm Theoretical Basis Documents. Before actual coding is done on a unit, subroutine, or module, the detailed design for that unit, subroutine, or module will be presented to a peer review group from the SDT using a walkthrough format. Coding (the implementation phase) will begin with the approval of the SDT Leader. The detailed design will be presented to the Science Team at the Detailed Design Review.

During the design phase, the SDT will begin to write the user’s guides and operations manuals. The software user's guide portion will provide instructions to the I-SIPS and IST operational staff on the use of the specific software. The software operational procedures manual portion will provide instructions to the I-SIPS and IST operational staff for operating, controlling, troubleshooting, and maintaining the specific software. The user's guide is a higher level section describing the proper use of the software, and any limitations or restrictions a user needs to know for the proper operation of the software. The maintenance manual section will be a compendium of the detailed design and the unit development information necessary for support of the particular software product release. The content will give an experienced programmer, not familiar with the developed software, sufficient information to perform maintenance operations and modifications. The user's guides and operations manuals for each SDS software system are combined as one deliverable document.

The user's guide/operational procedures manual documentation will be subjected to in-progress walkthroughs and reviews. The final versions of the user's guide/operational procedures manual will be presented at the Acceptance Review. As necessary, the Design Phase or portions of it will be ongoing until the software is completely designed and documented.

During this phase the SDT will begin development of the software assurance objectives and procedures.

6.1.1.1.3 Implementation and Testing Phase

The implementation and testing phase includes the implementation subphase and the integration and testing subphase. The implementation subphase comprises the coding and testing of the individual units, and will produce the Unit Development Folders (UDF). A UDF will be compiled for each software unit, subroutine, or module as identified by the SDT, and will be used to audit unit development activity. This folder will contain information such as unit status, unit detailed design description (e.g., pseudo-code, program definition language, etc.), special operating instructions, compiled code, unit test procedures, test results, software problem reports and changes, results of reviews (module walkthroughs), and notes. The contents of the UDF will be determined during the design phase.

Also during the implementation subphase, the software assurance and test documentation will be produced based on the NASA standards Assurance and Test Procedures Volume. Test specifications for the software builds will be generated, and included in the Assurance and Test Procedures. The test procedures for particular builds and integration level deliveries will be reviewed for acceptance by the SDT as part of the development process.

In addition to the test cases and data provided by the Science Team, the SDT will develop test data as needed. This will entail the assembly of actual aircraft flight, functional and comprehensive performance test data, along with laser altimeter simulation data to form a set of test cases. These test cases will be developed within the implementation subphase for support of the integration testing, and for acceptance testing in the subsequent phase activity.

The end of the implementation subphase will be marked by coded, compiled, and tested software units, subroutines, or modules, and data structures pertinent to the particular build being constructed.

The software integration and testing subphase comprises the integration of tested units into the programs for the particular build in progress and their testing. Integration is discussed in more detail in Section 6.1.3. This subphase produces the following documents and products as required by the SDT:

- Units, Build, and Integration Test Data Cases
- Test Reports (build and integration level)
- Version Description
- Units, Builds, and Integration-Level Software Products

The software test reports summarize the test results of the integration testing. These test reports are internal to the integration and testing, and are not substitutes for the acceptance testing and reporting activity. The Version Description (a roll-out of the Product Specification) will detail all the features found in the version being released, and all requirements which are to be satisfied by the build or integration level. The Version Description is specifically oriented to the preparation of the software and documentation for subsequent delivery to the Science Team. The user's guide/operational procedures documentation will be updated as necessary during this phase.

Acceptance test procedures and test data cases will be developed during the Implementation and Testing Phase. These test procedures, to demonstrate to the Science Team and ESDIS that the SDS system meets its requirements, will be included in the Assurance and Test Procedures.

The software implementation and testing ends with an internal review of the integration-level software delivery by the SDT to the SDT Leader. With the approval of the SDT Leader, the software transitions to the software Acceptance and Delivery Phase.

6.1.1.1.4 Acceptance and Delivery Phase

The software Acceptance and Delivery Phase will produce the following deliverables:

- Acceptance Test Data Cases
- Software Acceptance Test Reports
- Acceptance Level Software
- Software Product Delivery Package
- Data User's Handbook

Within this phase, the SDT performs acceptance tests on the integrated, to be delivered, software product. The acceptance tests are performed following the procedures and using the test data developed in the previous phase. The acceptance tests are performed on the ICESat SCF. The SDT will prepare acceptance test reports documenting the results of the tests. Software problems, errors, discrepancies, and anomalies are reported through the Discrepancy Report process. The software acceptance phase ends with the presentation of acceptance test results to the Science Team at the Acceptance Review.

Successful completion of the acceptance testing will authorize the developing of the software delivery packages, consisting of the software product, its associated product specification documents, test procedures and results, test data cases, and release description information. The planned contents of the delivery package are presented in Section 11. The transition from acceptance to delivery and installation occurs after Science Team approval.

The software package is delivered to the appropriate receiving organization for installation by the receiving team or the SDT. The installation involves the loading, compilation, assembly, and testing of the delivery software product as well as an examination of the associated documentation. The Software Acceptance and Delivery Phase ends with the successful demonstration of the delivered software product with the test data cases. The process culminates with an Operations Readiness Review, a presentation and demonstration of all software functions incorporated in the particular software product delivery.

Also during this phase, a data product description document (Data User's Handbook) will be produced which presents to the users descriptions of the data products generated by the GLAS SDS.

6.1.1.1.5 Sustaining Engineering and Operations Phase

Sustaining engineering and operations activities involve software maintenance, performance analysis, and operational support. These activities will continue as long as the software is in use. Any changes to the software will be implemented according to GLAS SDS configuration management procedures using the Discrepancy Report or Engineering Change Proposal, as appropriate. Since all the proper documents will have been generated, any significant revisions will be properly tracked via configuration control, and amended documentation generated as necessary.

6.1.1.2 Software Engineering Master Schedule

The products and reviews in the life-cycle are summarized in Table 6-1 "Software Engineering Master Schedule".

6.1.1.3 Software Computing Resource Requirements Estimation Techniques

This section will describe the techniques used to estimate the computing resource requirements for the GLAS SDS. These techniques are to be determined, but the resource requirements are relatable to the number of input/output parameters.

6.1.2 Prototyping

6.1.2.1 Purpose and Objectives

Prototyping will be an important part of the GLAS SDS development effort controlled by this Software Management Plan. It will be used in the earlier stages of the life cycle, the Requirements and Design Phases, as a means of insuring that the requirements are properly defined and complete, and to validate portions of the design which employ new technology, or which may be able to reuse existing code from closely related projects.

Table 6-1 Software Engineering Master Schedule

Life Cycle Phase: Subphase Activity	Deliverable Product	Review
Requirements: Concept and Initiation	Management Plan Volume: Software Management Plan†	SDT Leader
	Data Management Plan†	SDT Leader
Requirements: Requirements	Product Specification Volume: Requirements Document†*	Requirements/Architectural Design Review
Design: Architectural Design	Product Specification Volume Architectural Design Document†*	Requirements/Architectural Design Review
Design: Detailed Design	Product Specification Volume: Detailed Design Document†	Detailed Design Review
	Product Specification Volume: User's Guide/Operational Procedures Manual†*	SDT Leader
Implementation and Testing: Implementation	Assurance and Test Procedures Volume: Assurance and Test Procedures†*	SDT Leader
	Unit Development Folders§	SDT Leader
	Test Data Cases§	SDT Leader
	Management, Engineering, and Assurance Reports Volume: Test Reports§	SDT Leader
Implementation and Testing: Integration and Testing	Management, Engineering, and Assurance Reports Volume: Test Reports§	SDT Leader
	Product Specification Volume: Version Description†*	Science Team
	Units, Builds, and Integrated Software§†	SDT Leader
Acceptance and Delivery: Acceptance	Acceptance Test Data Cases†*	SDT Leader
	Management, Engineering, and Assurance Reports Volume: Acceptance Test Reports†*	Acceptance Review
Acceptance and Delivery: Delivery and Installation	GLAS SDS Delivery Package†	Operations Readiness Review
	Science Data User's Handbook	Science Team
Sustaining Engineering and Operations	Updated GLAS SDS Delivery Package†	Operations Readiness Review
All Phases	Management, Engineering, and Assurance Reports Volume: Performance/Status Reports# Discrepancy Reports# Engineering Change Proposals#	SDT Leader GLAS SDS Change Control Board GLAS SDS Change Control Board

Legend:
† - multiple editions planned based on preliminary or final versions, breakdown of SDS, or beta, engineering, launch, and updated releases
* - included as part of the software delivery package at beta, engineering, launch, and updated release
§ - number based on computer software design and implementation
# - as required or needed

6.1.2.2 Products and By-Products

Prototyping, as applied on the GLAS SDS development under the auspices of SDT, will be tracked through the use of a brief Prototyping Plan. The length and detail of the plan will be proportional to the complexity of the prototyping effort. Prior to beginning any prototyping, the following items will be addressed in the plan:

- Objective of prototype
- Statement of work
- Products definition
- Completion criteria
- Evaluation criteria
- Technical approach
- Resources required
- Schedule

6.1.2.3 Feasibility and Risks

Given the lead time in the software development process for the GLAS SDS prior to the scheduled beta version release, sufficient time is available to perform a prototyping activity. The experience of the SDT in supporting algorithm prototyping for previous satellite altimeter missions readily lends this capability for GLAS software algorithm analysis.

In the application of prototyping there is, however, a risk associated with the delay between the requirements development and the inception of the design phase. Routine management review of this process and the scope of its application should serve to mitigate this risk factor.

6.1.2.4 Description of Characteristics and Methods

Software developed to create the standard GLAS data products will involve discrete algorithms designed and implemented under the auspices of the Science Team. Prototyping will be used to evaluate the individual algorithms, and to determine the best method for grouping the algorithms into functional units.

The software developed to assess the GLAS instrument performance will consist of special purpose modules, specifically tuned to track the altimeter performance as it progresses from bread-boarding, through production and extensive ground testing, and finally into space where attentive monitoring will continue. Techniques already developed at GSFC and WFF for TOPEX/Poseidon, Mars Orbiter Laser Altimeter, the Altimeter Ice Database, etc., will be evaluated, through prototyping, for use on the GLAS instrument performance assessment task.

6.1.2.5 Analysis and Evaluation

The SDT Leader will sanction the prototyping, monitor its progress, and review the completion items and deliverables. The prototyping effort will, in general, follow the phases of the life cycle, but will neither generate the various reports nor involve as many reviews.

6.1.3 Integration

In the software development process, integration involves the collection or assembly of computer software units into a progressively more complete skeleton of the ultimate delivered system. The software units may in fact be program modules, subroutines, functions, or other subprogram structures, or physical entities either conforming to some lowest level of design organization, or to some function or software requirement that is testable. The integration test is the verification that this physical assembly of program code units complies with the design specifications and with the software requirements.

After the detailed design process has produced a mature design and prior to initiating code implementation, the integration definition and schedule will be formed. The integration activity will be performed as part of the code implementation process, and integration testing will be performed on a collection of deliverable units as part of the internal software development activities.

The GLAS SDS will be incrementally developed. Incremental development involves identifying groups of units to be integrated, integrating a group, testing the group as a package, and delivering the tested group. Tested groups are integrated together and tested as a new package. The scheme for these integration groups or segments will be based on algorithms, code requirements, and design specifications, and will be determined by the SDT. The application and accommodation of incremental code deliveries will be handled internal to the software development process.

The GLAS SDS will be managed subject to phased delivery in addition to incremental development. There are three ESDIS required deliveries - Beta, Engineering (V1), and Launch (V2). Each delivery will satisfy requirements as identified by the SDT during the requirements phase. With each delivery, the software will provide increasing capabilities. Any changes required subsequent to the V2 delivery resulting in a new release of the GLAS SDS will be designated as updated deliveries. Each delivery phase will be preceded by both integration testing and software product assurance testing under the auspices of the SDT.

Informal revisions to the GLAS SDS are assumed to be local to the software development effort. Informal revisions fix coding errors or modify code to meet the functionality of requirements and design. These revisions are managed as part of the routine life cycle activities and do not require a submission of an ECP unless these revisions significantly impact the code delivery schedule for integration and integration testing. Code revised in this manner will be integrated and tested according to the planned schedule. Code revised due to errors found during integration testing will be retested.

Changes required to the code that involve redefinition of algorithms, requirements, or design, even though internal to the software development process, are considered to be formal revisions within the GLAS SDS environment. These impacting revisions shall be detailed and submitted in an ECP for authorization by the SDT Leader. These changes may impact the integration definition or the integration schedule.

Revisions to software delivered to and accepted by the Science Team and ESDIS shall be considered formal. Detected errors or problems in delivered and accepted software products are expected to produce a Discrepancy Report which may result in an ECP. The ECP will be reviewed by the SDT and authorized by the GLAS SDS Change Control Board. The ECP process shall be able to accommodate problem reporting and change requests from outside the SDT. Formal revisions require both integration and software product acceptance tests to be performed prior to any delivery.

6.1.4 Engineering and Integration Support Environment

This section provides a general description of the engineering and integration support environment to be used for the GLAS SDS development. The following tools and application techniques are expected to be employed.

Documentation support will be provided for the production of various documents and reports. A standard set of tools will be chosen and adhered to as closely as possible. Any deviations from the standard set of tools must be approved by the SDT Leader. The standard set of tools will be chosen at the beginning of the requirements phase.

Currently, the document development platform is the Apple Macintosh personal computer. FrameMaker for the Macintosh is being used to create documents, with non-Macintosh personal computer text supplied by such tools as DOS WordPerfect and WordPerfect for Windows. Supplemental information involving tables, estimates, and other computational information may involve the application of Microsoft Excel for Macintosh. The primary tool for producing figures for documents is Macintosh Deneba Canvas. Currently, the project management tool is Microsoft Project. The documentation tools set will be used to support the reporting process across the life cycle phases and will include the production of reports as defined in Section 6.2.

During the requirements phase, the applicability of utilizing a Computer Assisted Software Engineering (CASE) tool in the software development activity will be investigated. If a CASE tool is employed, it will be used to produce various model diagrams such as context, data flow, and entity-relationship diagrams. It will also support a data dictionary and other document models such as events lists and process specifications. If a CASE tool is not utilized, the various diagrams, data dictionary, etc. will be produced using the standard tool set.

Supplemental tools that may be investigated for application in the software development activity, in addition to CASE tools, might be prototyping tools or automated prototype development platforms.

During the implementation and testing phase, the tools and techniques to be used for code implementation, integration, and integration testing will be based on the capabilities available on the ICESat SCF and on the GLAS IST. The process will employ personal computer word processing tools and UNIX operating system text editing tools such as the visual editor for code construction. Integration and integration testing are expected to use standard UNIX features such as shell script files, makefiles, and debugging tools. The ICESat SCF is specified to have the UNIX operating system architecture, and the standard high-level development languages FORTRAN-90 and C will be available for coding. Other language standards may additionally be incorporated for the GLAS SDS. UNIX extending or enhancing environments such as X Windows, Open Look, Open Windows, and OSF/Motif may become part of the development environment.

Pseudocode, protocode, or code generation tools may be investigated for application. Additionally, automated code testing platforms or applications might be used to support controlled testing and evaluation for the GLAS SDS. These tools will be considered during the requirements phase and added to the developmental tool complement used in the implementation and testing life cycle phase.

The Acceptance and Delivery Phase is expected to use the UNIX operating system, language, and utilities from the ICESat SCF and GLAS IST environments to support software product assurance through acceptance testing. Test platforms or applications may be considered for assurance testing activities. Network support applications, UNIX file transfer, and TCP/IP connectivity are expected to be used to deliver the software product and documentation to the DAAC facility and team. Additional computation, data evaluation, and instrument and software assessment support during this life cycle phase will be provided by the use of the standard spreadsheet tool.

The Sustaining Engineering and Operations Phase will use the UNIX environment and tools to support the GLAS SDS. In the event software patches or modifications are required for operational software products, all tools and techniques used in the original development will be applicable.

The incidence of changes, revisions, and updates to tools and support environments will be managed through a standard updating procedure. A new edition, patch, update, or release of a tool, operating system, language, or support utility will be independently evaluated apart from the development activity. Once the new tool has been suitably evaluated and accepted, an ECP covering the tool upgrade will be submitted to the SDT Leader for authorization. The system environment, the previous tool edition, and all associated files will be thoroughly backed-up to ensure the ability to return to the previous developmental and operational environment, should problems arise with the new tool or upgrade. These procedures ensure that the existing current applications, the development environment, and the operational environment will be updated in an orderly fashion, not spontaneously or in a random manner which might result in loss of data, development flow, or capability.

No special security or safety restrictions are planned for the development engineering and integration environment.

Implementation, acceptance, and application of tools and techniques may be accommodated beyond this management plan with the authorization and approval of the SDT Leader.

6.2 Products and Reports

As described in Section 6.1, the software will be developed and delivered in phases. As defined in this document, the GLAS SDS products are the I-SIPS Software and the IST Software. This section will also define reports produced by the SDT during development of the GLAS SDS products.

6.2.1 Baselining Process

This section provides the GLAS SDS definition of product baseline and the GLAS product baselining procedures. A GLAS product baseline is a product which is officially accepted by the GLAS SDS Development Team. A product is defined as the software that is delivered at the end of a phase of the GLAS SDS development. Since the GLAS SDS development will occur in phases, there will be a baseline defined and accepted for each phase.

The baselining process will consist of formal (Project level) reviews and informal internal reviews, walkthroughs, inspections, and configuration audits to ensure that the product fulfills all applicable requirements. Output data will be audited against their specifications. A product will be considered as baselined when it has been determined:

- that all applicable requirements are met including any changes, waivers, and deviations as approved through the Engineering Change Proposal process (the development phases and their requirements are determined during the Requirements Phase of the GLAS SDS life cycle plan); and
- that the software produces data with the correct contents, format, and size as specified by the Product Specification Volume documents for GLAS SDS.

The steps in the baselining process are:

- requirements for a product are audited at formal and informal reviews and walkthroughs to determine that all applicable requirements are included, as defined for the current delivery;
- the design is audited against the requirements to ensure all requirements are met (the audit is performed at informal and formal reviews and walkthroughs);
- the product output is audited against its specifications; and
- upon passing all steps in the process, the product is accepted as the baseline; if applicable, it is accepted by configuration management.

6.2.2 Product Specification Roll-Out Definition

The GLAS SDS development will occur during the following life cycle phases - Requirements, Design, Implementation and Test, and Acceptance and Delivery. The life cycle phases are defined in Section 6.1. The development will be documented in the Product Specification Volume documentation for the GLAS SDS. Section 6.1 defines the portions of the Product Specification Volume that will be completed for the GLAS SDS development activities; this section will define those portions that will be rolled-out into individual documents (i.e., a roll-out is the development of a particular NASA standards volume section into a stand-alone document).

Requirements will be established and the software requirements document will be produced during the software Requirements Phase of the life cycle. The software requirements document is a roll-out of the Product Specification Volume for the GLAS SDS.

During the architectural design portion of the life cycle's Design Phase, a top-level design will be produced and an architectural design document will be released. The architectural design document is a roll-out of the Product Specification Volume.

During the software detailed design process of the Design Phase of the life cycle, the detailed design of the software will be produced and the detailed design document will be completed. The detailed design document is a roll-out of the Product Specification Volume.

The software user's guide/operational procedures manual will begin during the software Design Phase, but will be completed during the software Acceptance And Delivery Phase of the life cycle. The software user's guide/ operational procedures manual is a roll-out of the Product Specification Volume for the GLAS SDS.

6.2.3 Assurance and Test Procedures Roll-Out Definition

During the software Implementation and Test Phase and during the software Acceptance and Delivery Phase, the GLAS SDS products will be assured through reviews, walkthroughs, inspections, and tests. The types of assurance and the organizations responsible for the assurance activities are defined in Section 8.0. The technical procedures used to assure the software will be defined in the Assurance and Test Procedures document. Only those sections applying to the assurance activities described in this plan will be included in the Assurance and Test Procedures Document. There are currently no planned roll-outs to the Assurance and Test Procedures Document. The Assurance and Test Procedures document will be completed during the software Implementation and Test Phase.

During the development process, several types of reports will be produced by the SDT as defined in Table 6-2 "SDT Reports". The contents of each report are defined in Appendix E.

The Management, Engineering, and Assurance Reports Volume provides a repository of all reports specified in this Plan and generated during the software development life cycle. This document can either contain all reports or contain pointers to each report's location. In either case, this document contains a brief description of each report type and an index of all reports generated.

6.3 Formal Reviews

For the GLAS SDS development, only those designated on the schedule in Table 6-1 are considered formal reviews. The SDT Leader will schedule the formal reviews and will ensure that the SDT has successfully completed the phase of software develop-

Table 6-2 SDT Reports

Performance/Status Report	Informs management about the performance or status of a work-area or a product and will be generated weekly or as needed. The report will also be used to record the conduct or status of any assurance or review activity such as a formal or informal review, walkthrough, or inspection or analysis of a product. For the assurance and review activities, this report is generated as defined in the Assurance and Test Procedures document.
Test Report	Provides the status of a test or a sequence of tests to management. This report is generated as defined in the Assurance and Test Procedures document.
Engineering Change Proposal (ECP)	States a suggested change to a product. This report will cover waivers and deviations, and also tracks the resolution of the change through any required new delivery. This report is generated as needed.
Discrepancy Report	States a discrepancy of a product or a product specification from applicable requirements, standards, and procedures. This report is generated as needed.

ment in question and is ready to present information/software to the Science Team. For each of the formal reviews, there actually may be more than one review scheduled. There may be a set of formal reviews for both the I-SIPS Software and the GLAS IST Software and for each build of a software system. All other GLAS SDS reviews are defined as informal. The informal reviews include the assurance reviews and walkthroughs that are described in Section 8.

6.3.1 Requirements/Architectural Design Review

At the Requirements/Architectural Design Review, the SDT will present the requirements and architectural design to the Science Team. This review will be conducted after the architectural design is complete, allowing the Science Team to approve/disapprove the requirements/architectural design prior to completion of the detailed design phase. At this review, the Science Team will be able to delete, modify, or add requirements and suggest changes to the architectural design.

6.3.2 Detailed Design Review

At the Detailed Design Review, the SDT will present the detailed design to the Science Team. This review will be conducted early in the implementation subphase so that the Science Team will be able to approve/disapprove the detailed design prior to completion of the software implementation. At the review, the Science Team will be able to suggest changes to the detailed design.

6.3.3 Acceptance Review

At the Acceptance Review, the SDT will present to the Science Team the results of the software acceptance testing. The SDT delivers final copies of the software delivery package including the user's guide/ operational procedures manual. At this review, the Science Team approves/disapproves delivery of the software. The Science Team may request additional acceptance testing to be performed. This review will be conducted shortly after the completion of acceptance testing.

6.3.4 Operations Readiness Review

At the Operations Readiness Review, the SDT demonstrates that the software has been successfully delivered and installed. The receiving operations team (I-SIPS Team or Instrument Operations Team) demonstrates that the software has been successfully tested and that the team training is complete. The result of this review is that the Science Team approves/disapproves the software for operations.

6.4 Interface Control Plan

The purpose of this sub-section is to identify the external interfaces to the GLAS Standard Data Software, and to present the approach to the management and control of these interfaces in the software development process.

A working definition of the term external interfaces is given as the points at which the software system under development meets and interacts with the external environment. The external environment means anything outside of the software under development including hardware, software, and people.

6.4.1 Technical Interfaces

The following external interfaces have been identified for the SDS: the required input and output data necessary to operate the software; the computer facilities on which the software will operate; and the people who will develop and execute the software.

The input and output data are the run-time control inputs, the input data products required for software system operation, and the output data products and processing reports produced as a result of the operation of the software. During system implementation, these interfaces will be supported by test data sets.

The I-SIPS Software will be installed and executed on the ICESat SCF. The IST Software will be installed and executed on the GLAS IST. The I-SIPS and IST Software will be implemented and tested on its respective host for operations.

The SDS will be executed and monitored by members of the I-SIPS Team and GLAS Instrument Operations Team. The GLAS Science Team will evaluate the standard data products generated by the I-SIPS Software output and will validate the contents. The GLAS Science Team will review the instrument health and the commanding performance data output by the IST Software. The standard data products are delivered to the DAAC for archival and availability to the science community. The IST Software will interface to the EOC for instrument operations including commanding.

The GLAS Science Team influences the development of the software system algorithms through the algorithm specifications, the analysis of calibration data, the development of science units conversion factors, and the production of formal test data sets. The GLAS Science Team influences the specification of the GLAS standard data products. The GLAS Science and Instrument Teams and the ESDIS influence the design and implementation of the IST software and its input/output.

The SDT will interface with the ESDIS when installing and utilizing the Toolkits. The SDT will report problems or request changes to the Toolkits via discrepancy reports or ECPs to the ESDIS.

The SDS development effort proposes no formal interface working group organization or structure. These responsibilities will be informally sustained through the normal working relationships and interactions of the GLAS Instrument and Science Teams working with the SDT Leader.

ESDIS Project influences are specified through ESDIS documentation. In particular, the DAAC configuration and the Toolkits will be developed and managed under Project-controlled documents such as information documents referenced in Section 2.3 [References 2.3a and 2.3b].

Table 6-3 "GLAS SDS Technical Interfaces" lists the identified technical interfaces.

Table 6-3 GLAS SDS Technical Interfaces

TECHNICAL INTERFACES	INTERFACE TYPE
Run-time Control Inputs	Input Data
GLAS Level 0 Data Products	Input Data
Ancillary Data	Input Data
GLAS Level 1 Data Products	Input/Output Data
GLAS Level 2 Data Products	Output Data
Processing Reports	Output Data
ICESat SCF	Hardware
GLAS IST	Hardware
DAAC	Hardware
EOC	Hardware
SDT	Human
I-SIPS Team	Human
GLAS Science Team	Human
GLAS Instrument Operations Team	Human
GLAS Instrument Team	Human
ESDIS Toolkits	Software

6.4.2 Interface Controls

The management and operations of the external interfaces will be controlled through various mechanisms within the GLAS investigation and through the software development process. The initial interface control will be directed through the Science Team documents and the GLAS Algorithm Theoretical Basis Documents. These documents influence the required inputs for processing, units conversions, and desired science and engineering outputs.

The major external interfaces for the Level 1 and Level 2 data products will be specified and controlled through the External Interface Requirements and the External Interface Design Specifications. These will be sections of the product specification requirements and design documents which will be prepared during the Requirements and Design life cycle phases. The interface specification for the ancillary data and for the IST input and output files will be described in the GLAS Science Data Management Plan, Information Document 2.3g.

Specification and control of the data product and report users interface will be accommodated through the GLAS Science Data User's Handbook. This document is data product oriented, and does not represent the formal application of the user's guide definition under the documentation standards [Reference 2.2c]. The operations interface will be managed and controlled through the Science Software User's Guide/Operational Procedures Manuals. These documents will describe the appropriate environmental and operational interfaces for the software products, along with the required control input specification and procedures. The GLAS Data User's Handbook and the Science Software User's Guide/Operational Procedures Manuals will be produced during the Design Phase of the software engineering life cycle.

Upon delivery and acceptance within the GLAS Science Team, the software and documents will be placed under GLAS configuration control and management. Any revisions to the software and documents under GLAS configuration and control will be considered formal revisions, and must be approved by the GLAS SDS Change Control Board.

With the Launch (V2) delivery and operations team acceptance, the GLAS SDS and supporting documents are placed under formal ESDIS Project change control. This delivery represents the ESDIS Project baseline. Any subsequent software and/or document revisions would be submitted through the formal Project engineering change proposal process.

The ICESat SCF is managed and controlled by the GLAS Science Team under the GLAS SCF Plan which discusses configuration specification and control. The ESDIS Toolkits are under the authority and control of the ESDIS Project. Toolkit documents are under the jurisdiction and change control authority of the Project, and are taken as superior requirements and constraints relative to the GLAS SDS.

The GLAS Instrument Support Terminal and its interfaces are under the control of the GLAS Science Team in concert with the ESDIS Project.

6.5 Training for Development Personnel Planning

This section addresses the specification and coordination of training required for SDT members. It is assumed that SDT members do not require training in the areas of routine documentation, project management, and software engineering and integration tools and techniques. However, enhanced and extended UNIX system capabilities such as OSF/Motif, newly accepted high-level programming languages, or new applications for development such as a CASE tool, code generation, or program testing platforms may necessitate personnel training. Project standards or related implementations such as the Toolkits may also require added development personnel training.

Whenever possible, necessary training will be managed on an in-house basis, employing video tape, audio tape, manuals, and self-instructional tools and capabilities. On-site instructed courses may be required to support development platforms and environments such as object oriented languages, if adopted as software product standards. EOS and ESDIS-related training offered at GSFC/Greenbelt will be attended by designated SDT members, as determined by the SDT Leader. If necessary, vendor-provided and commercially developed training programs may be utilized to support additional applications tools and development environments. These training programs would be handled on a per-request, as-needed basis.

Section 7

Sustaining Engineering and Operations Activities Plan

Sustaining engineering and operations activities will commence upon acceptance of the delivered V2 software. This section defines the operations activities and the sustaining engineering process. Additionally, this section will describe the types of product support provided with the delivered software.

7.1 Operations Activities and Sustaining Engineering Process

This section describes the planned operations activities for the I-SIPS and IST Software. The operations activities to be performed by the I-SIPS Team for the I-SIPS Software are: execute, monitor, and troubleshoot software, evaluate data products, perform data quality assessment and produce metadata, and deliver the products to the DAAC for archival. The major operations activities to be performed by the Instrument Operations Team for the IST Software are: monitor instrument health and perform instrument commanding.The I-SIPS Software will be executed on the ICESat SCF. IST support will occur at the GLAS Instrument Support Terminal. Operational procedures for normal and abnormal processing will be provided by the GLAS SDS Development Team (SDT).

Sustaining engineering is defined as the process of maintaining the delivered software throughout its lifetime. Sustaining engineering performed in response to an approved change request or as a scheduled maintenance activity will mirror development activities as described in this document. The requirements will be determined, and then the change or update will be designed, implemented, and tested. The change or update will be integrated into the current software and will be acceptance tested by the SDT. After acceptance by the SDT, the new software is delivered to the ICESat SCF or to the IST as appropriate where acceptance testing will be completed by the I-SIPS Team or GLAS Instrument Team. Upon being accepted, the software will be put into operations. Updated documentation will be delivered with the new or modified software. Version number and dates will be updated with each new release.

A change request can be initiated internally or externally to the SDT. Internally, the SDT will use an ECP form, which is described in Section 6; an example ECP form is presented in Table D-3 "GLAS Standard Data Software Engineering Change Proposal Form" on page -3. The ECP form includes areas for the initiation, reason, classification, priority, and description of the change; these are completed by the person requesting the change. Evaluation of the change and the approval disposition are included on the ECP form. The evaluation will include the resources to make the change and the feasibility of the change; the evaluation is completed by the SDT. The ECP is reviewed and authorized by the GLAS SDS Change Control Board. Upon approval, the ECP is implemented; a description of the implementation is included on the ECP form. Upon acceptance by the receiving team (either the I-SIPS or Instrument Operations Team), the change will become operational. The SDT will be able to handle change requests in any form since, for example, ESDIS may have their own type of change request initiation system. An external change request on a different form will be transferred to a GLAS SDS ECP form. The GLAS SDS ECP process will be capable of handling emergency change requests.

7.2 Product Support

The SDT will provide user and operator training, technical assistance, and documentation during delivery and transition to the I-SIPS Team and to the GLAS Instrument Team as discussed in Section 11. The SDT will perform software updates and other software maintenance activities as part of the sustaining engineering process. Updated documentation will be delivered with new or changed software; training on the new or changed software will be provided as needed. Training for new members of the I-SIPS or GLAS Instrument Operations Teams will be handled internally by that team.

Section 8

Assurance Plan

This Assurance Plan section addresses the planned activities to assure that the GLAS Standard Data Software and its associated documentation satisfy the requirements as specified in the software requirements documentation, and conform to applicable NASA, GSFC, ESDIS, and GLAS criteria. This plan is developed under the documentation standards of the NASA software engineering program.

Due to the small size of the project (fewer than 20 people), the GLAS SDS Development Team (SDT) will not employ an independent assurance organization. However, the SDT has established the following procedures to ensure that activities discussed in this plan are carried out and that good quality, reliable software is delivered.

1) Only the SDT Leader may authorize delivery of software and documentation.
2) Members of the SDT will inspect units as the unit coding is complete.
3) There will be an informal assurance activity audit trail. Inspection and review checklists and unit test results will be stored in Unit Development Folders (UDFs), and formal test reports will be stored in the Reports Volume.
4) Review of documentation will be accomplished through SDT reviews and project management review.
5) There will be an SDT member, the Assurance Monitor, in charge of the assurance effort. The Assurance Monitor, will have the responsibility to ensure that assurance activities as planned in this document are completed and that results are recorded as required. The Assurance Monitor will be responsible for preparing the Assurance and Test Procedures Document, and will collect and report the metrics defined in this plan.

The following sections define the plans for the different types of assurance activities to be employed by the SDT. For each assurance type, the approach and specific activities, methods, and products are defined. Metrics to be collected during each activity are also discussed.

8.1 Quality Assurance

8.1.1 Approach and Activities

Quality assurance ensures the conformance of the software and documents to the ESDIS Project standards, the GLAS Science Software Management Plan, and other applicable standards. Quality assurance activities will measure the degree of conformity or nonconformity of software and documents to the identified standards. Quality assurance will verify that the software development is ready to transition from one life cycle phase to another. Prior to initiating any quality assurance activity, the appropriate standards and plans to be followed will be identified. At a minimum, this Plan and ESDIS standards where possible will be used. The quality assurance activities include any formal ESDIS and GLAS Project reviews and informal SDT QA reviews. Formal project review requirements will be defined by the cognizant organization (ESDIS or GLAS). The informal QA review requirements are defined in this document.

Prior to any formal reviews, all documentation and software that are required to be delivered or presented at the review will be evaluated by the Assurance Monitor for adherence to standards (specified by the cognizant organization). The Assurance Monitor will provide evaluation results to the SDT Leader. The SDT Leader is the authority to release the software/documentation to the GLAS Science or Instrument Teams and to the ESDIS Project. The GLAS Science Team will approve any documentation before its delivery to the ESDIS Project.

There are two types of informal QA reviews - peer and group. Peer QA reviews are reviews of the completed units, to assure adherence to coding standards and the detailed design. The peer QA reviews will occur simultaneously with code inspections (see section 8.2.2). These reviews are performed by members of the development team (but not the member who coded the unit). Group QA reviews will be centered around design and document progress, and will allow general discussion and agreement on the direction of a particular document or area of design. Group QA reviews ensure that the development is ready to move from the current phase to the next phase. The group QA reviews involve the pertinent members of the development team (depending on the topic).

Any nonconformity to the standards detected during the quality assurance activities will result in one of the following actions:

- the nonconformity will be corrected and the product re-evaluated or,
- upon the approval of the SDT Leader, the nonconformity will be documented in a waiver and allowed to stand, or
- the nonconformity will cause the standard to be modified (through an Engineering Change Proposal (ECP), if necessary).

8.1.2 Methods and Techniques

For formal reviews, the development team will prepare the required materials and evaluate them for conformity to standards. The SDT Leader will determine whether the materials are ready for the review. Formal review procedures, objectives, and reporting will be defined in the Assurance and Test Procedures document.

Peer QA reviews of code will be performed against a checklist. The checklist is planned to be a compilation of specific items to check, as well as references to appropriate standards. The checklist will be created by the Assurance Monitor during the development of the Assurance and Test Procedures Document. The reviewer will receive the UDF (with the review checklist) for the unit to be reviewed from the Assurance Monitor. The reviewer returns the UDF with the completed checklist to the Assurance Monitor, who will return the unit to the implementer for corrections if necessary.

For group QA reviews, documents or software will be circulated (either paper copies or electronically) among the participants prior to the review by the Assurance Monitor. At the review, the Assurance Monitor will compile the review comments, and after the review will provide the comments to the appropriate team member(s) who will take action as required by the comments.

Any additional procedures for these activities will be defined in the Assurance and Test Procedures document.

8.1.3 Products

Checklists will be completed by the peer QA reviewer and stored in the reviewed unit's UDF. Reports will be generated for formal reviews by the Assurance Monitor. If any other quality assurance reports are required, they will be produced in accordance with the templates and the guidelines for Management, Engineering, and Assurance Reports in the software documentation standards (reference 2.2c). Any reports required by either GLAS or ESDIS Project management elements will be listed and defined in the Assurance and Test Procedures Document.

The metrics for quality assurance are:

1) the number passed vs. the number of assurance activities planned (the Assurance Monitor will develop a plan of assurance activities from the detailed design)
2) the number of discrepancies found vs. the number resolved
3) the types of discrepancies by class

Metric 1 is an indicator of how well the software design or the implemented code adheres to the identified standards; metric 2 is a measure of the maintainability of the code and the reliability of the design; and metric 3 may uncover design problems, if the same types of discrepancies are repeating. The Assurance Monitor will determine the discrepancy classes during the development of the Acceptance and Test Procedures Document. The Assurance Monitor will track these metrics on a biweekly schedule beginning with the detailed design phase.

8.2 Verification

8.2.1 Approach and Activities

Verification will demonstrate that the to-be-delivered software satisfies the requirements as defined in the GLAS software requirements documents, and that it is built and it functions as specified in the detailed design documentation. Verification activities will be performed by members of the SDT under the guidance of the Assurance Monitor. All software requirements must be verified by using one or more of the activities listed below:

Code inspections - Each completed unit will be examined to determine any discrepancies between the code and the detailed design specification, and to determine if there are any coding errors. The unit will be reviewed by an SDT member other than the one who coded the unit. The detailed design documentation will define the software units to be implemented. Additionally, code inspection can be used to verify requirements that can not be demonstrated during testing.

Verification Walkthroughs - A presentation of the design or code to date, with discussion centered on results of actions taken since the previous walkthrough and on the future plans for the software design or code. The walkthroughs will be conducted with the SDT or a subset of the SDT as applicable. At verification walkthroughs of design, the participants will meet to determine that the requirements are being incorporated into the detailed design; at verification walkthroughs of code, the participants will determine whether the code is complying with the detailed design.

Testing - Testing will ensure that the software fulfills the detailed design and the software requirements. Testing is briefly described in Section 4.2.1. Test planning will accommodate the various builds within the integration and testing phase, and will accommodate the incremental deliveries and delivery phases expected for the software and documentation products. The test procedures will accommodate the Beta, Engineering (V1), and Launch (V2) deliveries, and any update releases. There will be three types of testing performed by the SDT - unit, integration, and acceptance. The testing types and their purpose are defined inTable 8-1.

Table 8-1 The GLAS SDT Test Types

Test Type	Purpose
Unit	Verifies that the program unit behaves as specified in the detailed design documentation.
Integration	Builds module (or subsystem) by adding in program units. Verifies that the implemented module (or subsystem) matches the one specified in the detailed design documentation.
Acceptance	Verifies that the ready-to-be-delivered software satisfies all the requirements as stated in or derived from the software requirements documentation

Analysis of output - Determines algorithm performance at various stages of processing. This technique provides verification of complex algorithms by comparing results achieved through independent computations with those achieved from actual program execution.

Demonstration - The performance of the system is observed within a controlled environment. This allows the verification of some system functions, such as alarms, by actual observation.

Repeated usage - Repeated execution of the system in a controlled environment to determine reliability and functionality.

Discrepancies found during verification will either:

- be corrected by the implementer or,
- be documented with a waiver, upon approval by the SDT Leader or,
- will cause the requirements or design to change (documented with an ECP, if necessary).

8.2.2 Methods and Techniques

Code inspections are generally a line-by-line examination of the software, and will be performed by members of the SDT. After completion of unit testing (described below), the implementer will deliver the unit and its UDF to the Assurance Monitor. The Assurance Monitor will assign the unit to another member of the development team to perform the code inspection. The code inspection will ensure that

- the completed unit complies with the detailed design
- all equations and branches were tested during unit testing; any branches not verified by testing must be inspected by the reviewer to ensure that the branches are coded correctly and can be expected to yield the correct results.
- all successful tests are recorded in the UDF, and test results have been saved
- the process for any design changes is initiated
- any failed test that could not be fixed is recorded

The code inspection will occur simultaneously with the quality assurance peer review (see section 8.1.1). The reviewer will inspect the unit, using criteria for both types of assurance activities. A checklist will be developed by the Assurance Monitor for the reviewer to use as a review standard. This checklist will cover both the quality assurance and the verification criteria for passing review/inspection. The reviewer will complete the checklist, store it in the unit's UDF, and return the UDF to the Assurance Monitor, who will return the unit to the implementer to make any corrections. Otherwise, the unit will be marked as ready for integration into a module.

During a verification walkthrough, the author or implementer will present the detailed design or code in progress. The verification walkthrough participants have the opportunity to voice opinions on the direction or implementation of the detailed design. Any questions arising from a verification walkthrough that are not resolved at the walkthrough will be assigned to a SDT member(s) for resolution as an action item. The Assurance Monitor will compile the verification walkthrough comments and will ensure that any changes are implemented.

Unit testing is performed by the programmer after coding of a unit is completed and a clean compile is obtained. During unit testing, program logic will be checked completely - each branch, equation, and logic flow shall be verified. Any program logic that cannot be tested may be verified by code inspection.

Integration testing begins as units pass unit testing and become available for integration into modules. Integration testing will verify program calls and parameter passing. During integration testing, it is permissible for SDT members who have coded some of the units in the integration to perform the testing. Integration test procedures will be defined in the Assurance and Test Procedures document. The Assurance Monitor will participate in the integration testing to assure that all required documentation is completed and that the activity is progressing.

Acceptance testing is performed after the integration testing is complete for a software delivery package. Acceptance testing will verify that the completed software operates correctly and that it satisfies all requirements as defined in the software requirements documentation. The Assurance Monitor will lead and coordinate the acceptance testing. Acceptance test procedures will be defined in the Assurance and Test Procedures document. Testing by the SDT on the ICESat SCF and on the GLAS IST will be considered to be a reasonable emulation of the GLAS SDS operational systems. Formal acceptance testing of the GLAS SDS will be done by the I-SIPS Team and Instrument Operations Team.

During integration and acceptance testing, both the demonstration and repeated usage activities may be used to verify system reliability and functionality. The SDT may use analysis and code inspections to verify any detailed design elements or requirements that are too costly to verify through testing.

Any additional verification procedures will be defined in the Assurance and Test Procedures document.

8.2.3 Products

Though unit testing is defined to be informal, test procedures and a report must be written by the programmer and stored in the unit's UDF. The Acceptance and Test Procedures document will define what must be contained in the unit test procedures and report documentation.

The SDT will produce test reports during the integration and acceptance testing. All test reports produced will be in accordance with the templates and guidelines of the Management, Engineering, and Assurance Reports volume of the software documentation standards (reference 2.2c). Test results will also be documented in the Acceptance and Test Procedures document.

In addition to collecting the metrics listed in section 8.1.3, the verification activities will track the amount of time it takes to correct the code or the detailed design specifications. This metric provides a measure of the maintainability of the software. The Assurance Monitor will track these metrics on a biweekly schedule commencing with unit testing.

8.3 Quality Engineering Assurance

Quality engineering is the process of incorporating reliability, maintainability, and other quality factors into software products. By employing quality engineering techniques during software development, it is hoped to create a product that is reliable, maintainable, and portable. The GLAS SDS development effort will follow the life cycle plan as defined in Section 6. Quality engineering assurance activities will be incorporated as part of the quality assurance and verification activities as described in Sections 8.1. and 8.2.

8.4 Safety Assurance

Safety assurance entails the identification of software safety requirements, and ensuring that these requirements are satisfied. Software safety requirements are considered to include software performance with respect to hazards, faults, and other failure-related criteria.

No human or personal safety hazards have been identified as applicable to the GLAS SDS products.

Care must be taken to ensure that the SDS products supporting the IST do not inadvertently interfere with the instrument operations through the command and control process. The SDS will interface to the ESDIS Toolkits and it is expected that the toolkits will have the appropriate checks in place to prevent any interference with spacecraft or other off-limits operations. Additionally, during the detailed design phase, the SDT will develop a set of command rules which will provide guidelines and constraints for software and command developers in order to prevent any inadvertent interference with the spacecraft or instrument operations. It is the responsibility of the GLAS Instrument Operations Team to operate the IST facilities in a secure manner to prevent unauthorized use. As part of the assurance activities described in Sections 8.1 and 8.2, the software will be reviewed to determine that it complies with the command rules and restrictions, and tested to ensure that it traps command errors.

8.5 Security and Privacy Assurance

ESDIS is designed to be an information dissemination platform for Mission to Planet Earth. Therefore, privacy and proprietary considerations are not a concern for GLAS data and software. GLAS data products and software products are non-sensitive, and therefore security is not a key issue in the assurance planning process. There is, however, an integrity concern with any software product, its operational platform, and its input/output data. To protect the integrity of the software and data stored on the ICESat SCF and the IST, regular backup and archive procedures will be executed during the software development and mission operations. It is assumed that the ESDIS will have its own backup and archival system for DAAC operations.

Additionally, the configuration management techniques to be employed by the SDT will ensure the integrity of the GLAS SDS during development. See Section 10 for the GLAS SDS configuration management plan.

8.6 Certification and Validation

Certification is defined as the assurance process of confirming that the delivered software products are capable of meeting the requirements and other criteria in the actual operational environment. For the GLAS SDS, certification is done by the receiving organizations - the I-SIPS Team and the Instrument Operations Team.

Validation is the process of ensuring that the software actually produces the correct results, i.e., proves that correct requirements were provided to the development team. The validation process will be done by the GLAS Science and Instrument Teams.

Section 9

Risk Management Plan

Potential risks affecting the GLAS Standard Data Software development are identified in this section. The plans to monitor and minimize these risks are described.

9.1 Risk Assessment and Evaluation Process

The identification, evaluation, and minimization of the risks discussed in this section are based on past experience and lessons learned from other software development efforts. The overall plan to minimize risk is to follow the software development life cycle plan (discussed in Section 6), and thereby adequately define the requirements and develop a thorough, detailed design.

9.2 Technical Risks

The requirements and specifications for the GLAS SDS will come from the GLAS Science and Instrument Teams. The science algorithm information will be compiled into the GLAS Algorithm Theoretical Basis Documents by the GLAS Science Team. There is a risk that the GLAS SDS Development Team (SDT) may not understand how to apply the theories and algorithms discussed in the GLAS Algorithm Theoretical Basis Documents. To minimize this risk, algorithm workshops and reviews will be held with the GLAS Science Team, to discuss the SDT interpretation of the theories and algorithms. Additionally, the SDT may prototype the science algorithms and provide data to the GLAS Science Team for evaluation of the algorithms and their implementation. Test data from simulators and aircraft will be available for the prototyping activity in order that realistic results are obtained.

The GLAS SDS will interface with and utilize ESDIS Toolkits. Past experience with similarly described toolkits shows a high risk that the interface to the toolkit will be cumbersome or difficult to understand. To lower this risk, it is planned to review the documentation on the Toolkits early and to prototype a few cases of the utilization of the Toolkits. Further lowering the risk is, that by the time the SDS interfaces to the Toolkits are in development, the Toolkits will have been used by earlier ESDIS projects and therefore be in a refined form.

9.3 Safety Risks

The Instrument Operations Team through the Instrument Support Terminal (IST) will build and deliver commands to be uplinked to the spacecraft for the GLAS instrument. There is a risk that a command could adversely affect the GLAS instrument, the spacecraft, or another system onboard the spacecraft. To minimize this risk, the Instrument Operations Team will conform to a set of command rules which meet both spacecraft requirements and instrument requirements for GLAS instrument commanding. The command rules will be produced from constraints set by the ESDIS Project and the GLAS Instrument Team. A set of command restraints will also be imposed which will be adhered to by the Instrument Operations Team. The ESDIS Project's Flight Operations System should have command validation software which prevents any invalid GLAS generated commands from being sent to or accepted by the spacecraft. The GLAS instrument and the other systems should be built with onboard protection against unallowable commands.

9.4 Security Risks

With all software and hardware systems, there is a chance of loss of data or software due to hardware system failure or natural disaster. To mitigate the effect of either of these occurrences, regular backup procedures will be implemented. Additionally, copies of the software and data will be stored off-site.

As the GLAS data are not classified there is no security risk associated with data distribution.

9.5 Resource Risks

The GLAS SDS development activity will rely on NASA civil service employees and contractor employees. There is an inherent risk that when these contracts come up for renewal, the incumbent is not selected and a new contractor takes over. This could cause the work to get behind as new personnel are trained and brought up to speed on the details of the GLAS SDS development. However, this risk appears minimal because, historically, the new contractor will hire a high percentage of the personnel from the incumbent. To further minimize the risk, detailed documentation of the system will be available to the new contractor.

The GLAS SDS development is planned to occur on the ICESat SCF and the IST to the maximum extent possible. There is a risk that the ICESat SCF or the IST may not be available for software development activities. However, the effects of this risk are minimal as the development can occur on similar systems.

9.6 Schedule Risks

There is a risk that the planned schedules will not be met. To lower this risk, trackable tasks will be identified and project management tools will be used to track progress of these tasks in relation to the planned schedule. Milestones will be set according to required Project reviews and software delivery dates. The schedule will be built with a 3 to 6 month cushion in each phase to allow for any schedule slips.

9.7 Cost Risks

The risk of cost overruns is closely related to how well the work stays on schedule. When the work falls behind schedule, costs can go up either because it is taking longer to complete the task or because more manpower is assigned to get the task done on time. Funding will be established or released on a year-to-year basis. By monitoring the schedule and taking corrective action as necessary, the risk of cost overruns is minimized.

Section 10

Configuration Management Plan

10.1 Configuration Management Process Overview

Configuration management is the title given to the software management process sometimes referred to as change control or configuration control and management. The broader term configuration management is applicable because it encompasses not only change control but problem/failure reporting and resolution as well. Configuration management is a key process in the software development activities and is closely coupled with the software assurance processes.

The application of configuration management processes serves to identify guidelines and procedures for the establishment of software product versions and document editions and to control those software product versions and document editions. It includes the processes for software and documentation revision and update initiation, evaluation, review, disposition, and tracking. It accommodates a control form, the Engineering Change Proposal (ECP), that is applied throughout the revision process. For software or document anomalies, discrepancies, failures, or problems, it establishes a discrepancy reporting and tracking mechanism.

The configuration management process is applicable to the GLAS SDS and its documentation. While some mechanisms associated with configuration management will be employed from the beginning of the document and software composition processes, formal configuration management practices pertain to the baselined software and documentation products.

For the GLAS SDS, the software and several supporting documents will be delivered as a set known as the delivery package. Delivery packages correspond to the Project-required Beta, Engineering (V1), and Launch (V2) deliveries. The delivery software packages targeted for these delivery milestones will constitute the baseline.

Once a baseline has been established, configuration management practices become fully engaged. Changes, revisions, adaptations, modifications, enhancements, redesigns, etc., may be either required or desired for a baselined document or software product. These changes are proposed through the ECP process. The revisions or updates are proposed along with suggested implementation considerations, and are submitted for evaluation. An analysis and an evaluation are applied to the change proposal, a disposition and a resolution are identified, the change is accepted (or rejected), and is forwarded for implementation, testing, acceptance and/or tracking activities as required for the product and its maturity level. Figure 10-1 "GLAS Standard Data Software Configuration Management Processes" depicts the development process in conjunction with the configuration management process.The progression from the Beta version of the software toward the V1 version is considered to be evolutionary and is not subjected to change control or problem/failure reporting processes. However, in the development cycles from the Beta version to the V1 version, if

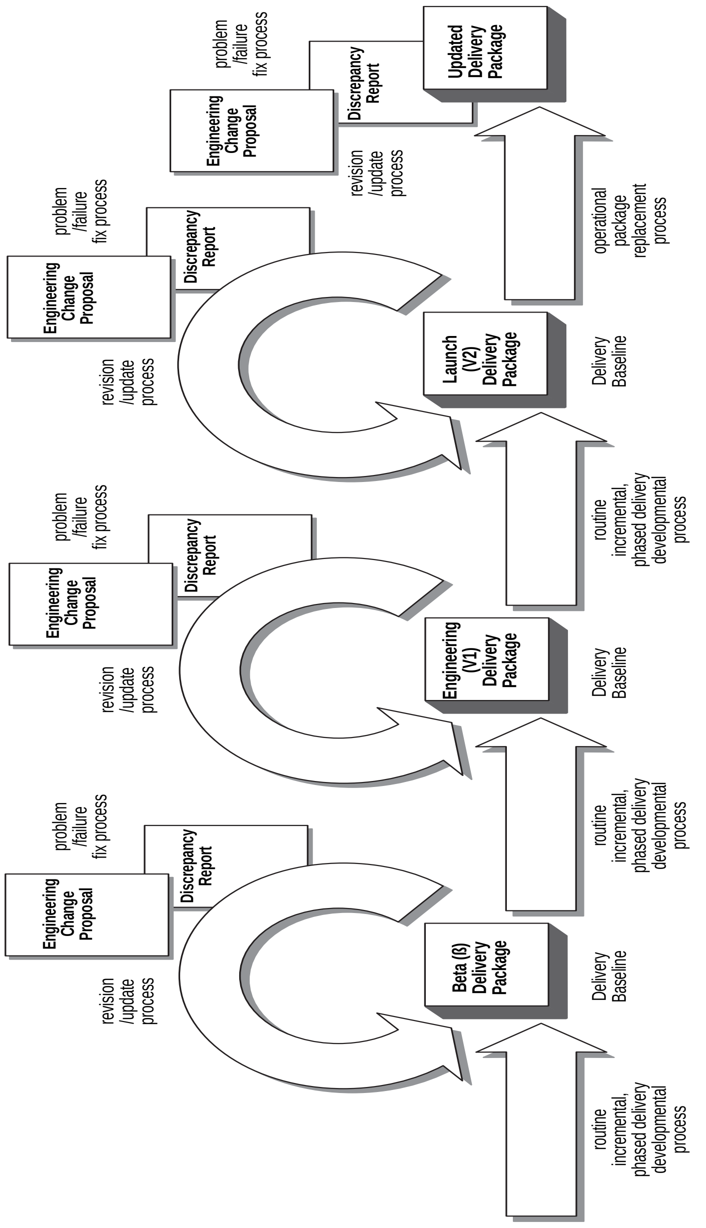

routine incremental, phased delivery developmental process
Delivery Baseline
Beta (ß) Delivery Package
revision /update process
Engineering Change Proposal
Discrepancy Report
problem /failure fix process
routine incremental, phased delivery developmental process
Delivery Baseline
Engineering (V1) Delivery Package
revision /update process
Engineering Change Proposal
Discrepancy Report
problem /failure fix process
routine incremental, phased delivery developmental process
Delivery Baseline
Launch (V2) Delivery Package
revision /update process
Engineering Change Proposal
Discrepancy Report
problem /failure fix process
operational package replacement process
Updated Delivery Package
revision /update process
Engineering Change Proposal
Discrepancy Report
problem /failure fix process

Figure 10-1 GLAS Standard Data Software Configuration Management Processes

significant revisions or updates are required or desired, and impact the baselined design or implementation processes, these changes shall be submitted as an ECP.

If a failure or discrepancy is detected, the configuration management process known as discrepancy reporting and corrective action is employed. This process of reporting, assessing the problem, and performing corrective actions is managed through the ECP process.

10.2 Configuration Control Activities

10.2.1 Configuration Identification

The configuration identification of a GLAS SDS unit consists of a software identification number and a documentation header. The software identification number will uniquely identify the software unit and will imply the hierarchical order of the unit within the overall product structure. The documentation header, incorporated into the software unit, is built in compliance with the SDT programming standard and guidelines. This header will contain the key required elements for configuration identification and management including the unique software identification number, the software unit date and time, and the version or revision number.

The software identification numbers will be determined and assigned by the SDT. The development evolution of a software unit will be further maintained through the use of the version/revision date, time, and number fields in the documentation header. The version/revision number field form shall be employed as per the SDT programming standards and guidelines. This number shall be revised to uniquely reflect each delivery, whether it be a baseline, build, unit, full system, or modification.

The documentation header shall contain and maintain the revision history of each software unit. The identification numbers shall be recorded in a record log to allow tracking and verification of the software configuration.

GLAS SDS documents will be assigned a unique identification number based on the document hierarchy depicted in Figure 1-1 "GLAS SDS Documentation Tree". A document is further identified by its version, date and a configuration status page. A configuration status page is contained within the document and lists any changes made to the document since it was first baselined. This page is updated to reflect any new releases of the document. An example of the configuration status page is shown in Table D-1 "Document Configuration Status Example" in Appendix D.

10.2.2 Configuration Change Control

The assigned responsibilities and activities within the configuration change control process for the SDT Leader and designated members of the SDT and the GLAS SDS Change Control Board are as follows.

The SDT Leader and SDT will:

- establish and assign the identification number for any GLAS SDS unit and document;
- create and maintain the GLAS SDS and document record log for configuration status accounting, and produce configuration status accounting reports as required by the SDT Leader;
- receive ECPs and Discrepancy Reports, assign unique proposal and report numbers, and log the proposals and reports into the configuration management log; monitor, and record the disposition of the ECPs and Discrepancy Reports throughout the SDT and the GLAS SDS Change Control Board evalu-

ation, analysis, review, authorization, and implementation steps; notify the organization originating the proposal or report of the outcome or disposition;

- review all submitted ECPs and Discrepancy Reports for content, merit, accuracy, and applicability; evaluate and analyze the impact, costs, resource demands, and schedule conflicts and implications associated with the change implementation, as well as the cost associated with a failure to accommodate the change or problem fix; and propose or evaluate a suggested approach or plan of action to implement the revision or correction;
- approve or reject the proposal or report for submission to the change control board. Approved ECPs and Discrepancy Reports will be forwarded to the change control board if there is to be a change in the requirements, or a change in the delivered products, or any change in the delivered version. Minor fixes such as coding bugs in the Beta, V1 or V2 versions will be handled locally;
- receive the authorized or rejected proposal or report from the change control board, review the board's analysis and suggestions, and direct the SDT in the implementation and resolution activities as required to institute the change;
- perform any software or document changes and fixes as authorized by the GLAS SDS Change Control Board;
- receive and review the implementation, action completion status from the development team, accept the implementation, revisions, etc., and authorize notification of the resolution to the originating organization;
- operate, maintain, fail-safe, manage, and distribute all electronic and paper records associated with the configuration management process;
- perform composition, update, distribution, maintenance, and revision of all GLAS SDS document products, to issue document updates, and produce Document Change Notices as required; and
- produce, as required, configuration status accounting reports to authenticate the configuration management processes.

The GLAS SDS Change Control Board will:

- receive, review, evaluate, and authorize or reject Engineering Change Proposals or Discrepancy Reports submitted through the SDT Leader; and
- audit the implementation, revision, resolution, and disposition of the authorized change reports to assure directions are carried out, and that reporting parties are properly notified of the report or proposal disposition.

10.2.2.1 Controlled Storage and Release Management

GLAS SDS baselined software products and documentation will be stored in designated controlled directory and file space allocated on the personal computers and workstations identified as the ICESat SCF. The controlled spaces for the storage of the baselined documents and software shall be accessible and maintainable by designated members of the SDT. The controlled spaces will also be accessible by designated members of the GLAS Science and Instrument Teams. The SDT will provide for the off-line back-up of the controlled document and software spaces.

Deliveries and releases authorized by the SDT Leader, either for internal use or Project distribution, will be delivered from controlled storage space to the designated recipient. Deliveries may be accomplished digitally through file transfer or electronic mail enclosures supported across the standard Ethernet access arrangement.

The primary security concern of the configuration management process is ensuring the integrity of the delivered baseline and authorized, revised products. ClearCase, a configuration management tool, will be employed to implement and manage the controlled directory and file space. Fail-safe/back-up activities shall protect the product integrity throughout the life cycle and the operational mission.

Access, userids, passwords, and directory space information will be protected throughout the configuration management processes. No other special access restrictions, privacy, or security measures are required or planned other than those already indicated as supportive of configuration integrity. All operations will be performed in accordance with GSFC and ESDIS security guidelines and requirements.

10.2.2.2 Change Control Flow

The following discussion details the flow of change requests and discrepancy report processes as presented in Figure 10-2 "GLAS Standard Data Software Engineering Change Proposal Flow" and Figure 10-3 "GLAS Standard Data Software Discrepancy Reporting Flow", relative to making a change to baselined software and documents.

Any cognizant GSFC, ESDIS Project, or GLAS investigation team, member, organizational element or group may initiate a change request or report a discrepancy in the software or documentation. The initiator will provide a description of the suggested change or the discrepancy. If available, supplemental information provided by the initiator will include implementation or solution proposals. The proposal or report is then delivered to the SDT Leader. The proposal or report will be transferred to a standard GLAS SDS Engineering Change Proposal or Discrepancy Report form if it is not already on one.

The proposal or report is forwarded to the appropriate members of the SDT for initial review and assessment, to determine whether the proposal or report is valid. If the Discrepancy Report is considered valid and a change to a software product or documentation is required, an ECP is issued. If an ECP is validated, then a solution is determined, the costs and resources associated with making the change are estimated, and the consequences of not making the change are evaluated. The ECP is then forwarded to the GLAS SDS Change Control Board for disposition.

If an ECP is rejected by the GLAS SDS Change Control Board, the initiator is notified by the SDT. If an ECP is authorized by the GLAS SDS Change Control Board, the SDT will implement and acceptance test the change and update any associated documentation. Upon approval by the SDT Leader, the new software and/or documentation is delivered to the appropriate operations team for their acceptance testing.

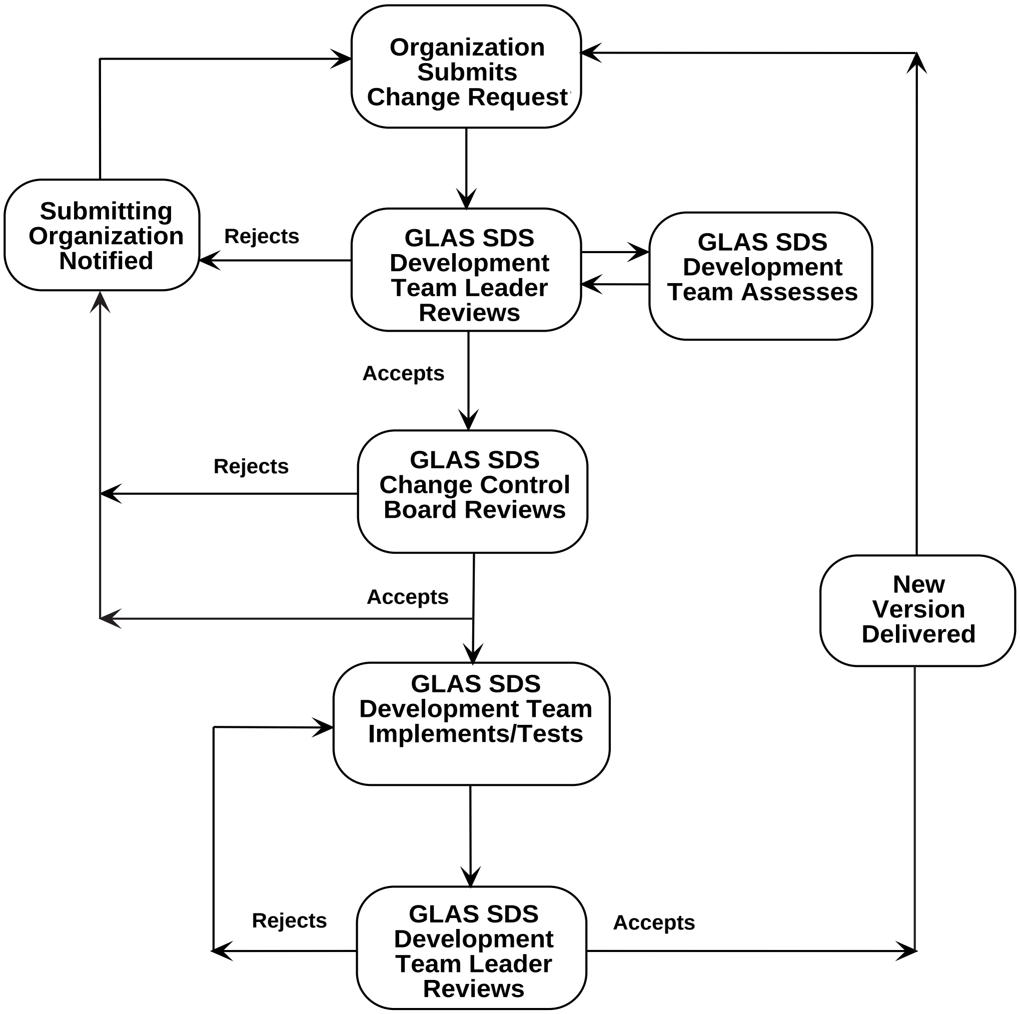

Organization Submits Change Request
Submitting Organization Notified
Rejects
GLAS SDS Development Team Leader Reviews
GLAS SDS Development Team Assesses
Accepts
Rejects
GLAS SDS Change Control Board Reviews
Accepts
New Version Delivered
GLAS SDS Development Team Implements/Tests
Rejects
GLAS SDS Development Team Leader Reviews
Accepts

Figure 10-2 GLAS Standard Data Software Engineering Change Proposal Flow

10.2.2.3 Change Documentation

The change documentation consists of the ECP and/or the Discrepancy Report. The guidelines for the contents of these reports and proposal documents was presented in Section 6.2.4. Table D-2 "GLAS Standard Data Software Discrepancy Report Form" and Table D-3 "GLAS Standard Data Software Engineering Change Proposal Form" in Appendix D contain the suggested structure and format for the GLAS SDS Discrepancy Report and ECP forms. These forms will be developed and implemented in accordance with the directions of this plan, subsequent procedures documents, and will conform to the standards presented in the Management, Engineering, and Assurance Reports volume templates and specifications.

10.2.2.4 Change Review Process

The GLAS SDS Change Control Board is the ultimate, official investigative body for change control approval of GLAS SDS and documents. It is responsible and accountable to the GLAS Science and Instrument Teams, the ESDIS Project, and GSFC man-

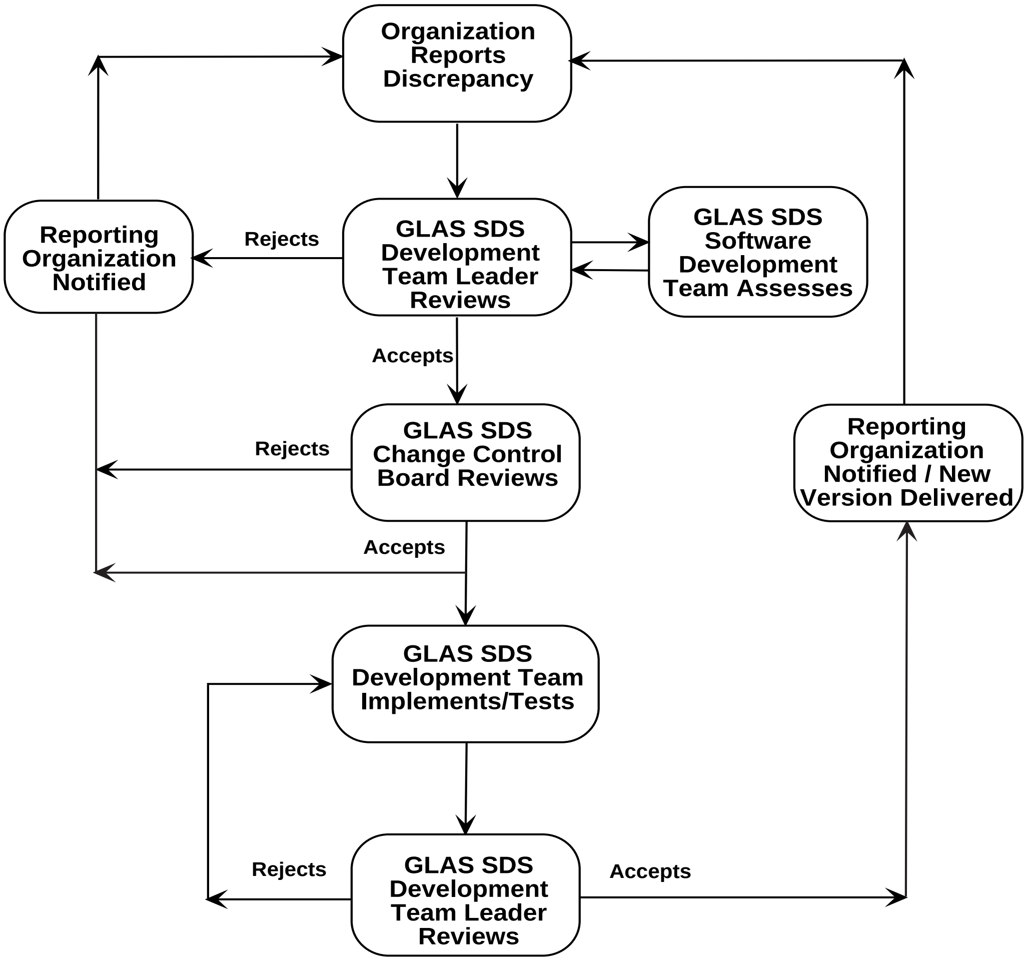

Organization Reports Discrepancy
Reporting Organization Notified
Rejects
GLAS SDS Development Team Leader Reviews
GLAS SDS Software Development Team Assesses
Accepts
Rejects
GLAS SDS Change Control Board Reviews
Reporting Organization Notified / New Version Delivered
Accepts
GLAS SDS Development Team Implements/Tests
Rejects
GLAS SDS Development Team Leader Reviews
Accepts

Figure 10-3 GLAS Standard Data Software Discrepancy Reporting Flow

agement for the operation and discharge of its assigned functions. It must evaluate and authorize or reject an ECP or a Discrepancy Report based on the proposal or report contents and any accompanying presentation materials and recommendations made by the SDT. The GLAS SDS Change Control Board will request any required collaborative information or analysis, and will solicit the SDT to support its evaluation and authorization process. The suggested board composition includes the SDT Leader, the GLAS Science Team Leader, and representatives from the instrument team and the science team. Figure 10-4 "GLAS Standard Data Software Change Control Board" on page 10-8 depicts the suggested composition of the GLAS SDS Change Control Board. It is the responsibility of the SDT Leader to determine when sufficient or time-critical business has been accumulated to convene the Change Control Board. When necessary, the board will address discrepancy reports and ECP disposition at the regular quarterly science team meetings. The Change Control Board may use the network media capability of either electronic mail or file transfer facilities to determine and provide dispositions on any discrepancy reports or ECPs.

The SDT will review each ECP and Discrepancy Report delivered by the SDT Leader to determine feasibility, implementation method, and implementation costs. This

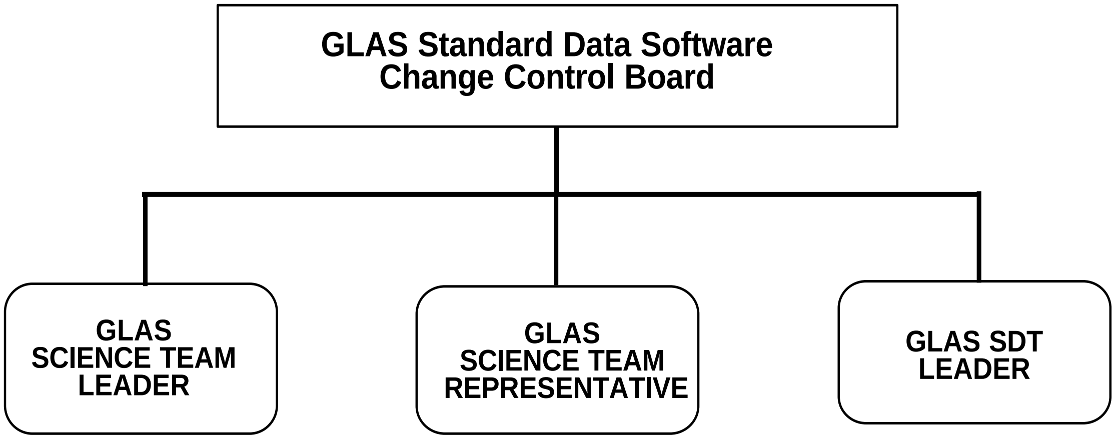

GLAS Standard Data Software Change Control Board
GLAS SCIENCE TEAM LEADER
GLAS SCIENCE TEAM REPRESENTATIVE
GLAS SDT LEADER

Figure 10-4 GLAS Standard Data Software Change Control Board

information is delivered to the GLAS SDS Change Control Board through the SDT Leader.

10.2.3 Configuration Status Accounting

A record log shall be maintained for the configuration status of both the GLAS SDS and its documentation. The purpose of these records is to show the current status of the contents of any GLAS SDS unit or document, as well as the historical evolution of these products. The log records will be a digital product maintained either as a table or as a data base product.

The following information, as applicable for each document or software unit, will be retained in the record log.

- unique identification number
- build, incremental delivery or phased delivery identification
  - name and description
  - date and time
- version and contents description
- archival information

A sample software product configuration status accounting report is provided in Table D-4 "Sample GLAS SDS Product Configuration Status Accounting Report" in Appendix D. A sample document product configuration status accounting report is provided in Table D-5 "Sample GLAS SDS Document Product Configuration Status Accounting Report" in Appendix D. Table D-1 "Document Configuration Status Example" in Appendix D provides a sample document configuration status page; this status page is designed to be included with each GLAS SDS document product.

Section 11

Delivery and Operational Transition Plan

11.1 Site Preparation Planning

11.1.1 Facility Planning

The GLAS Standard Data Software is composed of two systems - the I-SIPS Software and the IST Software as defined in Section 3. The I-SIPS Software will be delivered to and executed on the ICESat SCF, and the IST Software will be delivered to and executed on the GLAS IST. Additionally, the I-SIPS Software will be delivered to the DAAC for the ESDIS Project archive.The ICESat SCF will be developed to be adequate for the I-SIPS Software operations. A description of the ICESat SCF is contained in the GLAS *Science Computing Facility Plan*, Information Document 2.2f. The GLAS IST will be adequate for the IST support software operations. The GLAS IST is defined by the GLAS Instrument Team and the ESDIS Project.

11.1.2 Transition Planning

The following procedures will be performed to ensure that the sites (ICESat SCF and GLAS IST) are prepared for the delivery and transition of the SDS to the I-SIPS Team and the Instrument Operations Team.

- Coordinate delivery and transition schedule between the SDT and the I-SIPS Team and between the SDT and the Instrument Operations Team.
- The SDT points out any known discrepancies between the proposed and actual deliveries.
- The SDT ensures that all manuals and any other required software products are available.
- The SDT will provide technical assistance during the transition period.

The transition period is the period of time the operations team is learning how to use the software and is performing the acceptance tests on the software. The transition period begins when the software is delivered and ends when the software is accepted.

11.2 Delivery Planning

The following delivery and installation activities will be performed.

- The software, documentation, test procedures and test data, installation and operating instructions are delivered. Any additional elements required by ESDIS will be delivered.
- The I-SIPS Team will install and test the I-SIPS Software delivered to the ICESat SCF. The Instrument Operations Team will install and test the IST Software delivered to the GLAS IST.

- For a portion of the transition period, as needed, members of the SDT will be available at the delivery site for consultation and assistance. At other times, the SDT will be available at remote terminals.
- Upon successful installation and acceptance the software will become operational. At the ICESat SCF, the installation and acceptance period (synonymous with the transition period) is planned to last 2 months for each delivery. This will allow time for the operational personnel to become familiar with executing, monitoring, and troubleshooting the software.

The ESDIS Project requires three formal deliveries (Beta, V1, and V2) of the software. Table 11-1 "GLAS SDS Delivery Package Contents Description" lists items that will be included in each software delivery package. Any other items required by ESDIS will be included in the delivery package. The formal delivery schedule will be determined by the ESDIS Project. The scope of each of the three deliveries will be determined by the SDT during the requirements phase of the software development activity. The software will be delivered in electronic form; a paper copy and an electronic copy of each manual will also be delivered. Additional deliveries may be negotiated between the GLAS Science Team and the SDT.

Table 11-1 GLAS SDS Delivery Package Contents Description

Delivery Package Item	Contents/Composition
Data	Coefficients FIles Test Data Expected Test Results
Documentation	Software Version Description Software Requirements Document Software Architectural and Detailed Design Documents Software User’s Guide/Operational Procedures Document Data Management Plan Software Acceptance and Test Procedures Document Acceptance Test Reports
Source Code	Source Code Makefile Text Files Scripts

11.3 Data Conversion Planning

This section is not applicable.

11.4 User Training Planning

The following training will be provided during the delivery and transition period to the end users of the delivered software. End users are defined as those persons who will use the output of the software, i.e., data products and reports. This training applies to each planned delivery and to deliveries occurring as a result of any ECPs.

- Users are provided with the GLAS Science Software Data User's Handbook.
- Technical assistance regarding the data products and reports will be available from SDT members.
- User's guides or training for data product retrieval will be available from ESDIS.

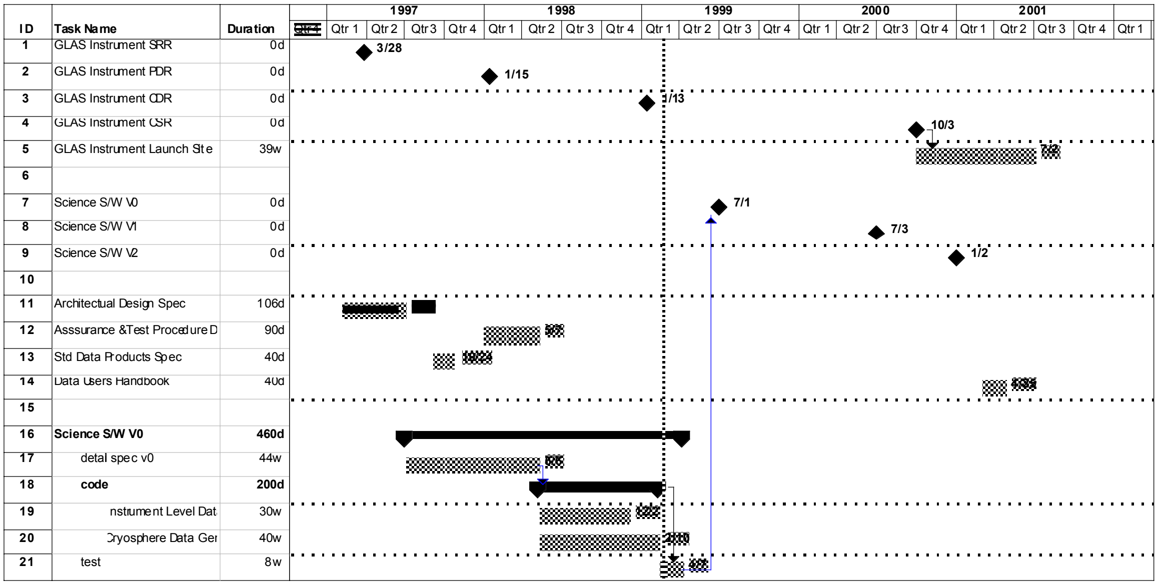
11.5 Operator Training Planning

The following training will be provided to the operations personnel - the I-SIPS Team and the Instrument Operations Team. (The operations personnel are defined as those persons who will execute, monitor, and troubleshoot the software.) This training applies to each planned delivery and to deliveries occurring as a result of any ECPs.

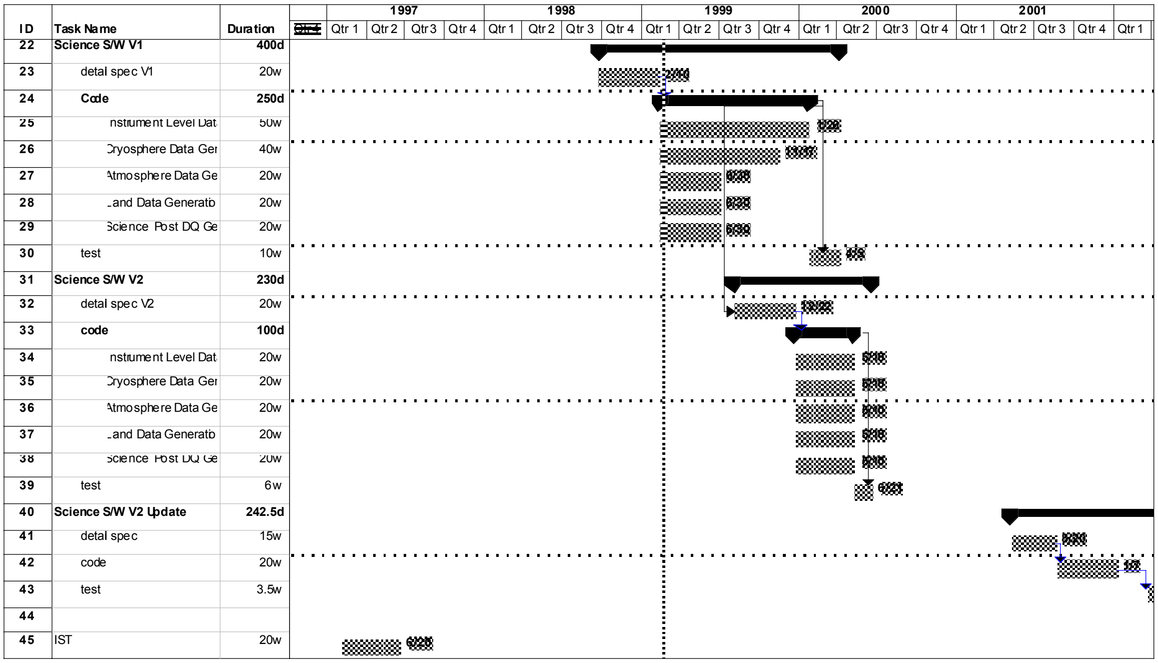
- Hands-on training will be administered during the transition period. There will be a SDT member available at the operations facility to provide the training.
- Operations personnel are provided with Software User's Guide/Operational Procedures Manuals.
- After the transition period, technical assistance from SDT members will be available at remote terminals.

Appendix A
WBS and Schedules

A sample schedule is shown in this Appendix. It will be revised as GLAS timelines become available.



ID	Task Name	Duration
1	GLAS Instrument SRR	0d
2	GLAS Instrument PDR	0d
3	GLAS Instrument CDR	0d
4	GLAS Instrument CSR	0d
5	GLAS Instrument Launch Site	39w
6		
7	Science S/W V0	0d
8	Science S/W V1	0d
9	Science S/W V2	0d
10		
11	Architectual Design Spec	106d
12	Asssurance &Test Procedure D	90d
13	Std Data Products Spec	40d
14	Data Users Handbook	40d
15		
16	**Science S/W V0**	**460d**
17	detal spec v0	44w
18	**code**	**200d**
19	nstrument Level Dat	30w
20	Cryosphere Data Ger	40w
21	test	8w



ID	Task Name	Duration
22	**Science S/W V1**	**400d**
23	detal spec V1	20w
24	**Code**	**250d**
25	nstrument Level Dat	50w
26	Cryosphere Data Ger	40w
27	Atmosphere Data Ge	20w
28	Land Data Generatio	20w
29	Science Post DQ Ge	20w
30	test	10w
31	**Science S/W V2**	**230d**
32	detal spec V2	20w
33	**code**	**100d**
34	nstrument Level Dat	20w
35	Cryosphere Data Ger	20w
36	Atmosphere Data Ge	20w
37	Land Data Generatio	20w
38	Science Post DQ Ge	20w
39	test	6w
40	**Science S/W V2 Update**	**242.5d**
41	detal spec	15w
42	code	20w
43	test	3.5w
44		
45	IST	20w

Appendix B

Funds and Budget Details

This information is not available for public dissemination.

Appendix C

SDS Documentation Tree and ID/Numbering

General Form:

GLAS-doctypacronym-serseqno

where GLAS-identifies the EOS ALT spacecraft instrument Geoscience Laser Altimeter System

doctypacronym- the three-character document type identification acronym (as per the document tree)

serseqno-identifies the document series-sequence number field which may take one of the following two forms depending on if the number represents a document, report, or a memo (using a "." (dot) if an additional sequence or sub-sequence identification is required)

1) GLAS-acn-vrln
for a GLAS document (vrln):

2) GLAS-acn-45yy.doy[.n]
for a GLAS memo or paper
(45yy.doy[.n]):

v	-volume ID	4	-reports volume ID
r	-roll-out no.	5	-memo ID
l	-level no.	yy	-year (last 2 digits)
n	-sequence no.	doy	-day of year no.
		n	-sequence no.

volume ID-identifies the NASA Software Engineering Volume Type:

1. identifies the Management Plan Volume
2. identifies the Product Specification Volume
3. identifies the Assurance and Test Procedures Volume
4. identifies the Management, Engineering, and Assurance Reports Volume

GLAS Standard Data Software Document Tree

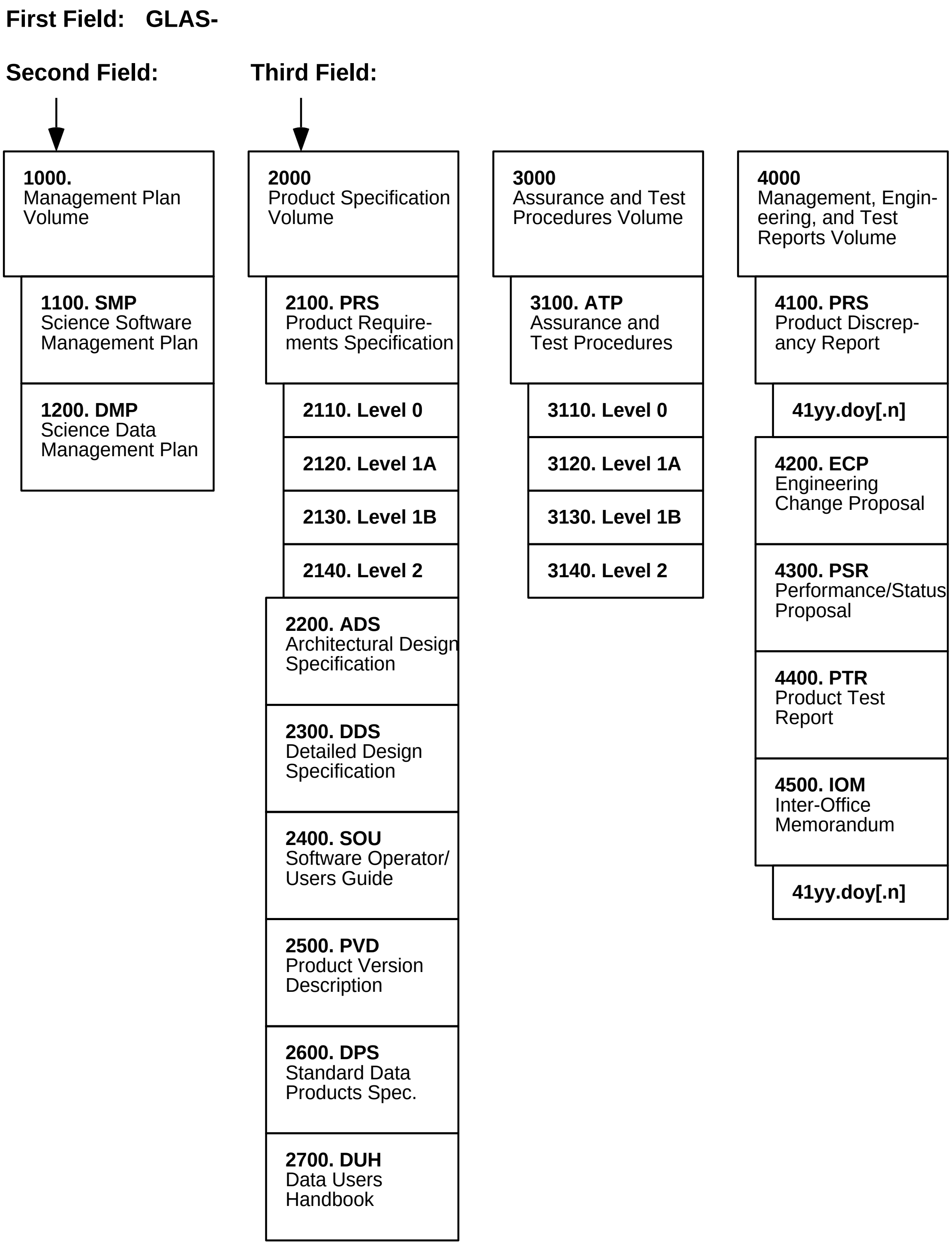

First Field: GLAS-
Second Field:
Third Field:
1000. Management Plan Volume
1100. SMP Science Software Management Plan
1200. DMP Science Data Management Plan
2000 Product Specification Volume
2100. PRS Product Requirements Specification
2110. Level 0
2120. Level 1A
2130. Level 1B
2140. Level 2
2200. ADS Architectural Design Specification
2300. DDS Detailed Design Specification
2400. SOU Software Operator/ Users Guide
2500. PVD Product Version Description
2600. DPS Standard Data Products Spec.
2700. DUH Data Users Handbook
3000 Assurance and Test Procedures Volume
3100. ATP Assurance and Test Procedures
3110. Level 0
3120. Level 1A
3130. Level 1B
3140. Level 2
4000 Management, Engineering, and Test Reports Volume
4100. PRS Product Discrepancy Report
41yy.doy[.n]
4200. ECP Engineering Change Proposal
4300. PSR Performance/Status Proposal
4400. PTR Product Test Report
4500. IOM Inter-Office Memorandum
41yy.doy[.n]

Figure C-1 GLAS Standard Data Software Document Tree

Appendix D

Configuration Management Forms and Reports

This appendix contains the configuration management forms and reports that are discussed in Section 10.

Table D-1 Document Configuration Status Example

Document Name:		**GEOSCIENCE LASER ALTIMETER SYSTEM** **Science Software Management Plan**	
Document Number:		**GLAS-SMP-1100**	
Page Configuration List			
Page Number	Issue	Page Number	Issue
cover page	Final	7-1 - 7-8	Final
signature page	Final	8-1 - 8-12	Final
configuration	Final	9-1 - 9-18	Final
i - iv	Final	10-1 - 10-8	Final
1-1 - 1-4	Final	11-1 - 11-6	Final
2-1 - 2-2	Final	12-1 - 12-20	Final
3-1 - 3-4	Final	13-1 - 13-16	Final
4-1 - 4-12	Final	14-1 - 14-4	Final
5-1 - 5-14	Final	15-1 - 15-2	Final
6-1 - 6-10	Final	16-1 - 16-32	Final
Document History			
Document No.	Status/Issue	Publication Date	Change Number
GLAS-SMP-1100	Preliminary	December 21, 1994	N/A
GLAS-SMP-1100	Preliminary	December 12, 1995	DCN-CN-1-02
GLAS-SMP-1100	Final	February 4, 1997	

Table D-2 GLAS Standard Data Software Discrepancy Report Form

Discrepancy Report Number:	
Originator Identification:	
Name:	
Organization:	
Address:	
Telephone:	
Product Identification:	
Name:	
Version Number:	
Environment Information:	
Life Cycle Phase Nonconformance Detected:	
Discrepancy Report Information:	
Title:	
Date:	
Nonconformance Type:	
Description:	
Proposed Solution Recommendation:	
Approval Authority Information:	
Criticality:	
Disposition:	
Resolution:	
Implementation Schedule:	
Corrective Action Designated Date/Version:	
Authority Signatures:	
Date Tested:	
Date of Closure:	

Table D-3 GLAS Standard Data Software Engineering Change Proposal Form

Engineering Change Proposal Number:	
Originator Identification:	
Name:	
Organization	
Address:	
Telephone:	
Product Identification:	
Name:	
Version Number:	
Environment Information:	
Engineering Change Proposal Information:	
Title:	
Date:	
Classification:	
Priority:	
Description of Proposed Change:	
Recommendation:	
Approval Authority Information:	
Disposition:	
Resolution:	
Implementation Schedule:	
Implementation Designated Date/Version:	
Authority Signatures:	
Date Tested:	
Date of Closure:	

Table D-4 Sample GLAS SDS Product Configuration Status Accounting Report

SOFTWARE PRODUCT CONFIGURATION STATUS ACCOUNTING REPORT							
Report Date:	05/12/96	Time:	12:10				
Software Product Identification:							
	Name:	GTRACK					
	Description:	GLAS Level 1A and Level 1B to Level 2 Ground Track Data Product Generation Program					
Software Delivery Identification						Date:	04/27/96
	Name:	Build 1.1				Time:	19:26
	Description:	Incremental build integration version through input processing including level 1A product open processing.					
Module Number	Module Name	Module Status	Module Version	Module Date	Spec. Date	Archive Location	Archive Date
h.1	inheader.h	prelim.	01.20	03/12/96	02/28/96	/usr/glas/slib/gtrack	04/27/94
0.0	main.c	draft	01.00	02/23/96	02/21/96	GLAS_XB_1:02	03/17/94
1.0	inproc.c	draft	01.02	02/26/96	02/22/96	GLAS_XB_1:02	03/17/94
1.1	setup.c	draft	01.10	02/28/96	02/21/96	GLAS_XB_1:02	03/17/94
1.1.1	bld_pths.c	prelim.	01.14	02/27/96	02/23/96	GLAS_XB_1:02	03/17/94
1.1.2	op_lev1a.c	draft	01.00	02/22/96	02/10/96	/usr/glas/slib/gtrack	04/27/94
1.1.3	op_lev1b.c	dummy	01.00	02/08/96	01/15/96	GLAS_XB_1:02	03/17/94
1.2	init_str.c	dummy	01.00	02/08/96	01/15/96	GLAS_XB_1:02	03/17/94
2.0	tproc.c	dummy	01.00	02/08/96	01/15/96	GLAS_XB_1:02	03/17/94
3.0	otproc.c	dummy	01.00	02/08/96	01/15/96	GLAS_XB_1:02	03/17/94
ESDIS Delivery Information:				Date:	05/03/96	Time:	13:25

Table D-5 Sample GLAS SDS Document Product Configuration Status Accounting Report

DOCUMENT PRODUCT CONFIGURATION STATUS ACCOUNTING REPORT				
Report Date:	09/16/95	Time:	15:03	
Document Product Identification:				
Document Name:		GEOSCIENCE LASER ALTIMETER SYSTEM - Science Software Management Plan		
Document Number:		GLAS-SMP-1100		
Document Delivery Identification:			Date:	09/12/95
Issue/Status:		in-progress Preliminary	Time:	10:44
Issue/Status Description:		Incremental PRELIMINARY document version through Section 17.0 less Section 17.2 contents		
Document Location:		Cabinet A, Drawer 3, Folder D-144		
ESDIS Delivery Information:	Date:	MM/DD/YY	Time:	HH:MM

Appendix E

SDS Development Report Contents

Performance/Status Report

- Identification of GLAS SDS activity or process.
- Author/Submitter/Report date.
- Accomplishments
- Differences between planned versus actual performance.
- Open items, problems.
- Recommendations

For assurance or review activity reporting, also include

- Identification of the assurance activity as specified in the Assurance and Test Procedures document (GLAS SDS assurance activity number).
- Identification of the product or process being evaluated including the version number and date as applicable.
- Identification of the responsible organization or person.
- Date and location of the assurance activity.
- Identification of the persons or organizations doing the evaluation.
- Summary of results - list any errors or discrepancies and recommendations made; report status of assurance activity; any ECPs or discrepancy reports.
- Action items and person(s) to whom assigned.
- Approval action and authority taken as a result of the activity.
- Date of follow-up, if necessary.

Test Report

- Identify the test as defined in the Assurance and Test Procedures document (GLAS SDS test number).
- Identify the product including the version number and date.
- Date of test.
- Tester(s)
- Test witnesses (if appropriate).
- Anomalies encountered and recovery procedures attempted.
- Test status and summary of results; attach test output as appropriate.
- Date of re-test, if necessary.

Engineering Change Proposal

- GLAS SDS ECP number.
- Originator information including name, address, phone, organization, and date.
- Product identification including name or title, version number and date, and environment information if applicable.
- Proposal information including title, date, classification (major or minor), priority (low, high, emergency), type (change, waiver, deviation), description of proposal, recommendation, date needed.
- Proposal analysis including rationale for change, waiver, or deviation; rationale for classification; required resources to make change (if applicable); effect on personnel, software, documentation, or other systems; impact on personnel, training or other systems if change, waiver, or deviation is not accepted; suggested resolution. Refer to any associated analysis.
- Change approval including disposition, resolution, implementation schedule, approval signature.
- Submitted and approved for change.
- Submitted after implementation for delivery/installation.
- Date of installation.

Discrepancy Report

- GLAS SDS Discrepancy Report number.
- Originator information including name, address, phone, organization and date.
- Identify product including name, version number and date, environment information, if applicable (e.g., hardware and operating system).
- Description of discrepancy.
- Recommendation; if known, include code, data, or documentation where corrective action must be taken.
- Approval including criticality, disposition, resolution, implementation schedule, new date/version of the item in which the corrective action will be included, approval signature, date tested, date closed.

Abbreviations & Acronyms

ATBD	GLAS Algorithm Theoretical Basis Document
BETA	EOSDIS GLAS GDS Software product Beta version release designation
CASE	Computer Assisted Software Engineering software development tool or platform
CDDIS	Crustal Dynamics Data and Information System
CDR	GLAS instrument or science software Critical Design Review
DAAC	EOSDIS Distributed Active Archive Center facility for sensor data record generation, and data product generation and distribution
ECP	Engineering Change Proposal report document
EDOS	EOSDIS Data and Operations System
EOC	EOS Operations Center
EOS	the NASA Earth Observing System Mission Program
EOSDIS	Earth Observing System Data and Information System
ESDIS	Earth Science Data and Information System
FY	government Fiscal Year designation
GLAS	Geoscience Laser Altimeter System instrument or investigation
GPS	Global Positioning System
GSFC	the NASA Goddard Space Flight Center at Greenbelt, Maryland
GSFC/WFF	the NASA Goddard Space Flight Center/Wallops Flight Facility at Wallops Island, Virginia
ICESat	Ice, Cloud, and Land Elevation Satellite
I-SIPS	ICESat-Science Investigator led Processing System
IST	GLAS Instrument Support Terminal facility and/or workstation
N/A	Not (/) Applicable
NASA	National Aeronautics and Space Administration
NOAA	National Oceanic and Atmospheric Administration
NSIDC	National Snow and Ice Data Center
PC	Personal Computer, generally used to refer to an Intel processor based, IBM or IBM-clone desktop computer
PDR	GLAS instrument or science software Preliminary Design Review
SCF	Science Computing Facility

SDS	Standard Data Software
SDT	SDS Development Team
SSMP	GLAS Science Software Management Plan document
TBD	to be determined, to be done, or to be developed
TCP/IP	Transmission Control Protocol/Internet Protocol network access standard/ protocol
UDF	Unit Development Folder
UNIX	UNIX operating system jointly developed by the AT&T Bell Laboratories and the University of California-Berkeley System Division
UTGLAS	University of Texas GLAS SCF node
V1	EOSDIS GLAS GDS Software product Engineering version release designation
V2	EOSDIS GLAS GDS Software product Launch version release designation
V&V	software assurance Verification and(&) Validation activity
WBS	Work Breakdown Structure
WFF	NASA Goddard Space Flight Center/Wallops Flight Facility at Wallops Island, Virginia

Glossary

aggregate — A collection, assemblage, or grouping of distinct data parts together to make a whole. It is generally used to indicate the grouping of GLAS data items, arrays, elements, and ESDIS parameters into a data record. For example, the collection of Level 1B ESDIS Data Parameters gathered to form a one-second Level 1B data record. It could be used to represent groupings of various GLAS data entities such as data items aggregated as an array, data items and arrays aggregated into a GLAS Data Element, GLAS Data Elements aggregated as an ESDIS Data Parameter, or ESDIS Data Parameters aggregated into a Data Product record.

array — An ordered arrangement of homogenous data items that may either be synchronous or asynchronous. An array of data items usually implies the ability to access individual data items or members of the array by an index. An array of GLAS data items might represent the three coordinates of a georeference location, a collection of values at a rate, or a collection of values describing an altimeter waveform.

element — Specifically, a GLAS Data Element. A GLAS Data Element is identified by a unique element number, and is composed of a data item or an array of items. A GLAS Data Element represents the decomposable unit of data contained in an ESDIS Data Parameter.

file — A collection of data stored as records and terminated by a physical or logical end-of-file (EOF) marker. The term usually applies to the collection within a storage device or storage media such as a disk file or a tape file. Loosely employed it is used to indicate a collection of GLAS data records without a standard label. For the Level 1A Data Product, the file would constitute the collection of one-second Level 1A data records generated in the SDPS working storage for a single pass.

header — A text and/or binary label or information record, record set, or block, prefacing a data record, record set, or a file. A header usually contains identifying or descriptive information, and may sometimes be embedded within a record rather than attached as a prefix.

granule — The designation for the smallest unit of distributable data for an instrument, experiment, or an investigation. It may indicate a pass, orbit, mapping or repeat cycle, or even a single record of data, However, when data is requested, this is the smallest quantity that can be retrieved. For a GLAS Level 1A data product, the pass (1/2 orbit) is the granule size.

header — A text and/or binary label or information record, record set, or block, prefacing a data record, record set, or a file. A header usually contains identifying or descriptive information, and may sometimes be embedded within a record rather than attached as a prefix.

item — Specifically, a data item. A discrete, non-decomposable unit of data, usually a single word or value in a data record, or a single value from a data array. The representation of a single GLAS data value within a data array or a GLAS Data Element.

label

The text and/or binary information records, record set, block, header, or headers prefacing a data file or linked to a data file sufficient to form a labeled data product. A standard label may imply a standard data product. A label may consist of a single header as well as multiple headers and markers depending on the defining authority.

Level 0

The level designation applied to an ESDIS data product that consists of raw instrument data, recorded at the original resolution, in time order, with any duplicate or redundant data packets removed.

Level 1A

The level designation applied to an ESDIS data product that consists of reconstructed, unprocessed Level 0 instrument data, recorded at the full resolution with time referenced data records, in time order. The data are annotated with ancillary information including radiometric and geometric calibration coefficients, and georeferencing parameter data (i.e., ephemeris data). The included, computed coefficients and parameter data have not however been applied to correct the Level 0 instrument data contents.

Level 1B

The level designation applied to an ESDIS data product that consists of Level 1A data that have been radiometrically corrected, processed from raw data into sensor data units, and have been geolocated according to applied georeferencing data.

Level 2

The level designation applied to an ESDIS data product that consists of derived geophysical data values, recorded at the same resolution, time order, and georeference location as the Level 1A or Level 1B data.

Level 3

The level designation applied to an ESDIS data product that consists of geophysical data values derived from Level 1 or Level 2 data, recorded at a temporally or spatially resampled resolution.

Level 4

The level designation applied to an ESDIS data product that consists of data from modeled output or resultant analysis of lower level data that are not directly derived by the GLAS instrument and supplemental sensors.

metadata

The textual information supplied as supplemental, descriptive information to a data product. It may consist of fixed or variable length records of ASCII data describing files, records, parameters, elements, items, formats, etc., that may serve as catalog, data base, keyword/value, header, or label data. This data may be parsable and searchable by some tool or utility program.

orbit revolution

The passage of time and spacecraft travel signifying a complete journey around a celestial or terrestrial body. For GLAS and the EOS LASER ALT spacecraft each orbit revolution count starts at the time when the spacecraft is on the equator traveling toward the North Pole, continues through the equator crossing as the spacecraft ground track moves toward the South Pole, and terminates when the spacecraft has reached the equator moving northward from the South Polar region.

packet

A data packet, referring to the basic aggregation of data values, usually raw data, as grouped in an instrument or flight computer, telemetry stream, or ground receiver system.

parameter

Specifically, an ESDIS Data Parameter. This is a defining, controlling, or constraining data unit associated with a EOS science community approved algorithm. It is identified by an ESDIS Parameter Number and Parameter Name. An ESDIS Data Parameter within the GLAS Data Product is composed of one or more GLAS Data Elements.

pass

A sub-segment of an orbit, it may consist of the ascending or descending portion of an orbit (e.g., a descending pass would consist of the ground track segment beginning with the northernmost point of travel through the following southernmost point of travel), or the segment above or below the equator (e.g., either the northern or southern hemisphere portion of the ground track on any orbit).

product

Specifically, the Data Product or the ESDIS Data Product. This is implicitly the labeled data product or the data product as produced by software on the ECS or SCF. A GLAS data product refers to the data file or record collection either prefaced with a product label or standard formatted data label or linked to a product label or standard formatted data label file. Loosely used, it may indicate a single pass file aggregation, or the entire set of product files contained in a data repository.

record

A specific organization or aggregate of data items. It represents the collection of ESDIS Data Parameters within a given time interval, such as a one-second data record. It is the first level decomposition of a product file.

repeat cycle

The time span or number of orbits elapsed when a later orbit ground track superimposes on (or repeats) an earlier orbit's ground track.

Standard Data Product

Specifically, a GLAS Standard Data Product. It represents an EOS LASER ALT/GLAS Data Product produced on the ECS for GLAS data product generation or within the GLAS Science Computing Facility using EOS science community approved algorithms. It is routinely produced and is intended to be archived in the ESDIS data repository for EOS user community-wide access and retrieval.

variable

Usually a reference in a computer program to a storage location.